1. Introduction: Java and Compilers for Java

The Java programming language is a "simple, object-oriented, network-savvy, interpreted, robust, secure, architecture neutral, portable, high-performance, multithreaded, dynamic language[1]"(cited from: [1] http://java.sun.com/java.sun.com/doc/Overviews/java/) It's a useful and powerful tool to program the Internet, intranets and any other distributed network. In order to make a Java program run on different platforms, a Java compiler generates bytecode based on the architecture of the Java Virtual Machine and the compiled code is interpreted on the particular platform. The quality of the compiled bytecode will directly affect the execution efficiency. This paper, which is based on a Master's project in Java compiler, is intended to describe the techniques for building a Java compiler. To simplify the problem, the compiler only compiles the one-class Java source code of either an application or an applet, and does not handle scope.

2. Design: the Design Issues

2.1 Components of the Compiler

The whole compiler consists of a lexical analizer, a YACC-based parser, symbol table manipulation, code generator, syntax-tree manipulation, constant-pool builder, method-call processor, and the target code generator.

The lexical analizer is basically a Lex program. The regular expressions used by Lex program is specified in ***The Java^{TM} Language Specification*** (*Version 1.0 Beta*). The analizer reads the Java source code, recognize the lower-level structures as *terminal symbols*, and returns these tokens to the paser.

The YACC-based parser reads the information provided by the lexical analizer, recognize higher-level structures as *nonterminal symbols*, form the parse tree, and generate the intermediate Java assembler code based on syntax-directed translation.

The symbol table is built in the parsing process. It's required that a variable must be declaraed before it's used. For a one-class source code, the symbol table keeps records of all names and the corresponding attributes, including the register number allocated to the variables.

Code generator generates the intermediate Java assembler code in accordance with the Java assembler code generated by `javap`. A self-specified protocol was used by the code generator to facilitate the process. The code are also optimized by the Code Optimizer which is a part of the Code Generator..

The syntax-tree is built by the parser in the parsing process. The main purpose is to help generate the intermediate Java assembler code. The tree node that contains a variable name always connects to the symbol table. Before a segment of code are generated, one or more tree nodes are referenced or a part of the tree is traversed to obtain information such as date type or to access the symbol table to get the register number allocated for a variable, and so on.

The constant-pool generator constructs the constant pool which is a part of the target Java bytecode. The data in the constant pool is conllected in the parsing process and the constant pool is constructed at the end of the parsing process.

The method-call processor recognize method calls and generates code for them. Since this is a simplified compiler, all method code are hard-coded. If the whole set of methods are to be handled, a library have to be built to contain the index to methods and the corresponding code for the methods.

The target code generator operates at the end of the compiling process. It translates the intermediate Java assembler code into bytecode and generate the `.class` file. For detailed information about the format of the `.class` file, please refer to ***The Java Virtual Machine Specification*** (*Release 1.0 Beta*).

2.2 Java Grammar

The java grammar for building the compiler is a context-free grammar, based on the specification of version 1.0 of the Java Language, but some modifications were made. The grammar notation specified in ***The Java^{TM} Language Specification*** (*Version 1.0 Beta*) was used to describe the working grammar. For detailed information, please refer to ***Appendix A: Java Grammar***. The grammer is translated by YACC with 11 'good' shift/reduce conflicts. In the compiler implementation, a very little simplification was made in order to illustrates the techniques and the conflicts was reduced to 6.

2.3 The LALR(1) Method

2.3.1 Intruduction to the LALR(1) Technique

An important part of the compiler is the parser which was written by using YACC. YACC uses the LALR(1) method to construct parsing tables. An LALR(1) parser can be built by first constructing an LR(1) parser, and then merging states. In particular, an LALR(1) parser is an LR(1) parser in which all states that differ only the lookahead components of the configurations are merged. In LR(k) parsing techniques, "L" means the input is scanned from left to right, the "R" stands for constructing a rightmost derivation in reverse, the "k" is the number of the lookahead symbols that are used to make decisions in parsing.

2.2.2 Reduce the Grammar Ambiguity

One problem confuses the YACC comes from the three definitions of the grammar specified in ***The JavaTM Language Specification*** (*Version 1.0 Beta*). These three definitions are: variable decalration, method declaration and constructor declaration. All three declarations need some modifiers and a modifier may apply to two or three declarations. In order to enable YACC to distinguish the three definitions in parsing, the modifiers are classified into following classes:

1) common modifiers that can apply to all three declarations:

```
VariableMethodConstructorModifier:    public
                                      private
                                      protected
```

2) modifiers that can apply to both varible and method declarations:

```
VariableMethodModifier:    static
                           final
```

3) modifiers that can apply only to variable declaration:

VariableOnlyModifier: transient
volatile

4) modifiers that can apply only to method declarations

MethodOnlyModifier: abstract
native
synchronized

Accordingly, the grammar rules for the three declarations are modified as follows:

VariableMethodConstructorDeclaration: ->VariableMethodModifier VariableMethodDeclaration
->VariableMethodDeclaration
-> ConstructorDeclaration

VariableMethodDeclaration -> VariableOnlyModifier Type VariableDeclarator ;
-> Type VariableOrMethodDeclaration
-> MethodOnlyModifier void MethodDeclarator $Throws_{opt}$ MethodBody
-> MethodOnlyModifier Type MethodDeclarator $Throws_{opt}$ MethodBody
-> void MethodDeclarator $Throws_{opt}$ MethodBody

VariableOrMethodDeclaration-> VariableDeclarator ;
-> MethodDeclarator $Throws_{opt}$ MethodBody

ConstructorDeclaration -> ConstructorDeclarator $Throws_{opt}$ ConstructorBody

As shown above, the grammar will directed the YACC program to look ahead step by step when the parser reads the input of the three declarations and thus eliminate the ambiguity and errors that could result from the grammar described in the Java specification.

2.4 Code Generation

Since the Java virtual machine is a stack-based machine, it's easy for the compiler to use the syntax-directed translation for the code generation. In syntax-directed translation, we parse the input token stream, build the parse tree and then traverse the tree as needed to evaluate the semantic rules at the parse-tree nodes. In the process of evaluating the semantic rules, code are generated, information is saved in a symbol table and error messages are issued whenever error occurs.

The compiler uses multipass analysis techniques. In the first pass (in the parsing process), most of the code -- code for expressions, data structure references, control structures and method calls -- are generated expcet the addresses of the jump instructions. In the second pass (after the parsing process), the code generated in the first pass are analized and the addresses of the jump instructions are determined. In this way, the intermediate assembler code is generated.

2.5 Code Optimization

The conpiler implements two optimization strategies:

1. Constant Expression (and part of sub constant expression) Evaluation at Compile Time
   For example, for the following expressions
   b = 8 * 9 + 10 -7 * (5+8); c = 8*9*b;

the compiler will generate the following code as if:

b = -9; c = 72*b;

2. Dead-Code Elimination

This optimization is done within the basic blocks based on live variable analysis. For example, for the following expressions

a = 5; a = 6;

the compiler only generates code for a = 6;

2.6 Register Allocation

In Java, register allocation occurs in method declarations. It differs between static methods and non-static methods. A static method cannot be overwritten, thus the key word `this` is not permitted to use. In such a method, registers are allocated from 0, and each local variable is assigned a different register number depending on the sequence of the variable declared. This is pretty straight forward. However, a non-static method can be overwritten and thus in such a method, register 0 is always reserved for `this` class. The first register that can be allocated to a local variable is register 1, not 0.

For example, in a static method like:

```
static void main(String argv[])
{
            int i, j, k;
            ...
}
```

Variable `argv` obtains register 0, `i` gets register 1, `j` get 2, and so on.

In a non-static method like:

```
void paint(Graphics g)
{
            Font f;
            float p;
            ...
}
```

Variable `g` will be assigned register 1, `f` gets register 2, and `p` has register 3. Register 0 is reserved for `this`.

2.7 Limitations

1) The compiler's symbol table does not handle scope, which means within a method common variable names in different scopes may cause errors.

2) There is no type-checking. The compiler can only detects syntax errors.

3) The compiler only handles following system-defined methods:

```
println(String);
println(String + number..);
```

if there are more than one numbers in the parameter list, each number must be separated by strings.

```
resize(int, int);
setColor(Color);
drawString(String, int, int);
```

```
setFont(Font);
fillRect(int, int, int, int);
drawRect(int, int, int, int);
setBackground(Color);
fillOval(int, int, int, int);
repaint();
```

The compiler handles the following system-defined field access:

```
java.lang.System.out
java.awt.Color.red
java.awt.Color.green
java.awt.Color.black
java.awt.Color.yellow
java.awt.Color.white
```

4) The optimizer can only handle basic data types (int, float) used in the testing programs. But it can handle all data types in the above method calls which the compiler is designed to be able to handle.

5) The optimizer may not handle the assignments in a `for` block, `while` block or `if` block properly because of the incomplete analysis based on the basic blocks. This means, if a variable is assigned a value inside those blocks and referenced outside those blocks, it may be deleted during the optimization. So, the following code segment may not be optimized correctly:

```
i = 0;
while (i < 100) {
   println("The value of i is: " + i);
   i = i + 10;
}
```

In the above code segment, the optimizer will delete the statement:

```
i = i + 10;
```

The best way is to re-write such loop by using for:

```
for (i = 0; i < 100; i = i + 10) { ...}
```

One way to avoid such optimization error, at current development stage, is to declare i as a field of the class, so that it's global to all the methods of the class. Another way is to avoid using optimization.

6) Only support postfix increment.

7) If a variable is initialized in declaration, never declare it with other variables. Never write: `int a, b=10, c;` ALWAYS write: `int a, c; int b=10;`

8) Maximum number of self-defined methods in a class: 100

9) Maximum number of constant pool entries: 2400

10) Maximum number of self-defined fields: 64

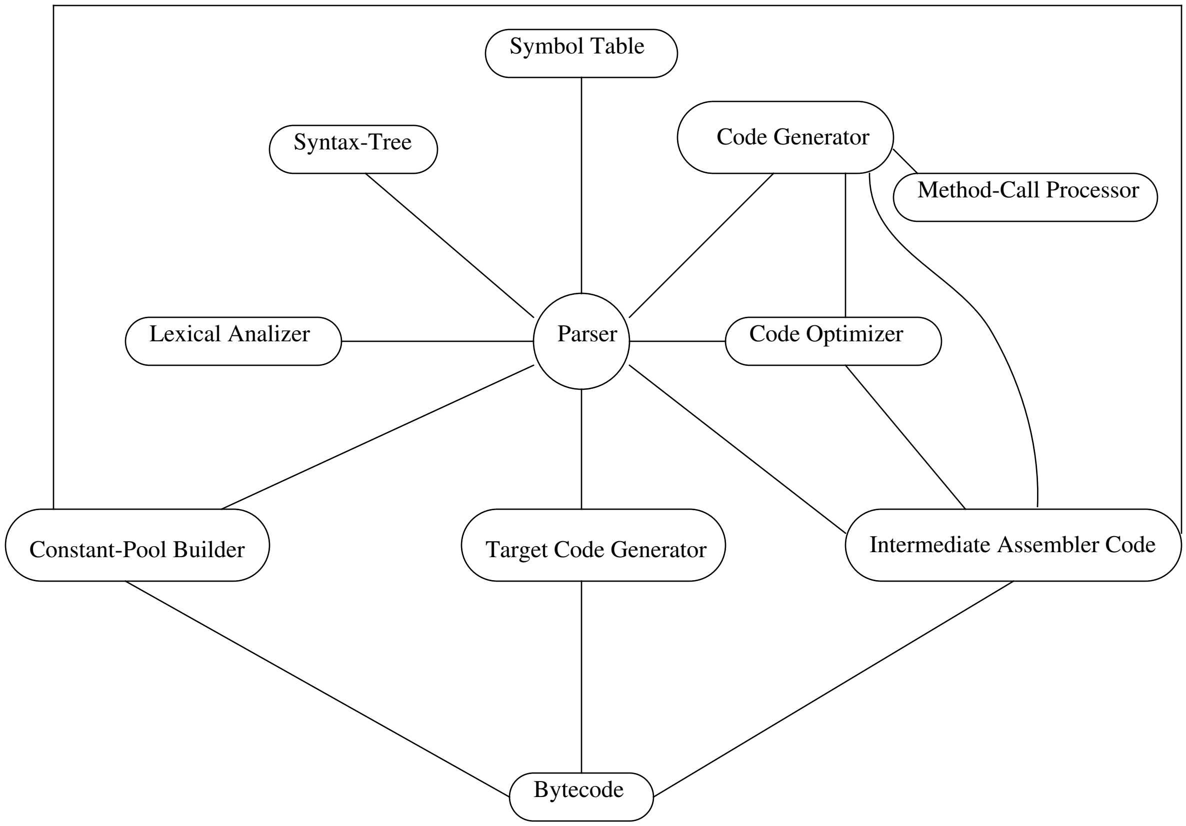
11) Maximum number of nested method calls: 10

12) Maximum number of nested control structure: 16

3. Implementation Details and the Issues

3.1 Structure of the Java Compiler

The structure of the compiler is shown as follows:



Structure of the Java Compiler

The compiler first generates the intermediate code in the parsing process, and optimizes the imtermediate code right after the parsing process. The optimized assembler code is stored in a temporary file. At this point, part of the constant pool is built. Then the compiler analysizes the intermediate code, adds constants for method calls, finishes the constant pool. Meanwhile, part of the target bytecode is generated in a memory buffer. At last, the constant pool is spooled to the disk and the bytecode in the buffer is appended to the constant pool. The target bytecode runnable file is generated in this way.

3.2 The Java Virtual Machine Architecture

For detailed information about the Java Virtual Machine, please refer to ***The Java Virtual Machine Specification*** (*Release 1.0 Beta*) or the web site at:

http://java.sun.com/java.sun.com/doc/

3.3 Data Structures

The major types are defined as follows:

1) Data Types

enum type_enum {j_int = 0, j_long, j_float, j_double, j_byte, j_short, j_char, j_boolean, j_ref};

2) Information about Each Type

```
struct type_info_t {
        int storage_class;              /* the symbol on the node is a class */
        enum type_enum type;            /* the data type of the symbol */
        int qualifier;                  /* not used */
        int function: 1;                /* the symbol on the node is a method name */
        int array: 1;                   /* the symbol on the node is an array type */
        int constant;                   /* the node contains a constant */
}
```

3) Structure of Symbol Table Nodes

```
struct SymbolTableNode {
                char symbol[MAXIDLEN+1];
                int  symtype : 1;       /* symbol type is either  symbol  or typedef */
                int  reserved;
                int  array;             /* if the symbol is of an array type */
                int  func;              /* not used */
                int  count;             /* not used */
                int  scope;             /* not used */
                int  local_cnt;         /* the register number assigned to the symbol */
                int  address;           /* not used */
                int  isField;           /* if the symbol is the field */
                enum type_enum  type;   /* date type of the symbol */
                char *strngvalue;
                struct SymbolTableNode *next;
};
```

4) Structures for the Syntax-Tree

```
struct node {
  char *symbol;                         /* symbol name */
  int num_children;                     /* number of children */
  struct node *children[Max_Children];  /* array of pointers to each child */
};

struct {
        struct node *tree_info;                 /* pointer to tree info node */
        struct SymbolTableNode *symptr;         /* pointer to a symbol table entry */
        struct type_info_t type_info;   /* type information of a node */
        struct list* truelist;          /* true list for the control statements */
        struct list* falselist;         /* false list for the control statements */
        struct list* nextlist;          /* next list for the control statements */
        float const_val;                /* constant value if the node has a constant */
}
```

5) Structures for the Constant-Pool

```
struct CONSTANT_Utf8_POOL {     /* used to represent constant string values */
      unsigned char  tag;       /* here, always CONSTANT_Utf8            */
```

```
    unsigned short length;     /* length of the actual string              */
    unsigned char  *bytes;     /* point to the actual string               */
};

struct NAMEANDTYPE_INDEX {        /* for method ref, field ref  */
    unsigned char tag;
    unsigned short nameIdx;    /* name index into the utf8 pool      */
    unsigned short sigIdx;     /* signature index into the utf8 pool */
};

struct CONSTANT_INDEX {
    unsigned char  tag;
    unsigned short index1;           /* name, class, name and type, string */
    unsigned short index2;           /* signature index; default:          */
    unsigned char  floatBytes[64];   /* IEEE 754 standard later...         */
    unsigned short oldIdx;           /* 'line number' for re-arranging CP  */
};
```

3.4 What can be handled by the compiler?

In order to illustrate the basic compiling process, the compiler compiles one-class file at a time and there is no type-checking. When compiling a class, the compiler is able to hanlde:

1) Data Types: integer, float, one-dimentional integer array, one-dimentional float array, some of system-defined date types such as String, Graphics, Font.
2) Statements: local variable declarations, empty statement, if statement, for statement, return statement, assignment, post-increment, method call, allocation expression
3) eveluate expressions, some field-access (only for system-defined fields, e.g out, Color), reognized operators are '+', '-', '*', '/', '%', '<', '<=', '>', '>=', '==', '!='

3.5 The Assembler Bytecode -- The Intermediate Code Generation

Most of the intermediate assembler code is generated in the parsing process. The global variable `bytecnt` records the address for each instruction, and is reset for each method. The main functions that generates assembler code are:

```
void GenPushInt(int n);
void GenLDFloat(float fval);
void GenArrayOp(char *ins, int size, int local_cnt) ;
void gen(char *instruction, int offset, int localCounter);
```

When the parser is scanning within a method, the intermediate assembler code is generated in the buffer `buf and fieldBuf`. After finishing scannig the method, the contents of `buf and fieldBuf` are reorganized and spooled to a temporary file. The prototype of a method is kept in `mthdName`. At last, the `mthdname` and the temporary file is assembled together to make the final assembler code file. This final process is implemented by function :

```
void MakeIt(char *fn, char *name)
```

where `fn` is the source file name, and `name` is the class name.

3.5.1 `gen()` *protocol*

Function `gen` has a set of protocol to generate different format of instructions. The first argument `instriction` always contains the instruction name. The second argument `offset` usually is the length of the whole instruction line in bytes including the instruction and its operands. The third argument `localCounter` usually is the register number assigned to a variable.

If both the `offset` and `localCounter` is not negative, then the `instrction` should have a register number as its operand. In such a case, a underline '_' is used to connect the `instruction` and the register number.

If the `offset` is negative, then the default offset is 3 bytes and the `localCounter` represents the operand of the `instruction`. If the instruction line is not 3 bytes, it's adjusted elsewhere.

If the `localCounter` is negative and the `offset` is not negative, then the `instruction` has no operand.

For some special instructions, they are recognized by the `gen()` function and treated specially. For example, the `iinc` instruction has two operands and the whole instruction length is 3 bytes. in this case, both the offset and the localCounter act as the two operands.

3.5.2 `GenPushInt`

When the parser sees an integer constant, it calls the function `GenPushInt` to push the interger constant on to the operand stack of the Java Virtual Machine. `GenPushInt` then choose the appropriate push instruction according to the value of the constant integer and calls `gen()` function to generate the real code.

3.5.3 `GenLDFloat`

When the parser sees a float constant, it calls the function `GenLDFloat` to generate instructions for loading the float constant from the constant pool and pushing it on to the operand stack. It calls `gen()` to generate the real code.

3.5.4 `GenArrayOP`

This function is used to load contents from an array variable. It calls `gen()` to generate the real code.

3.6 The Constant Pool Construction

Constant pool is an important commponent of the class file. For detailed information about the constant pool, please refer to ***The Java Virtual Machine Specification*** (*Release 1.0 Beta*).

3.6.1 General Mechanism

The information for constant pool is collected during parsing process, and is classified into three separate buffers: a Utf8 buffer, a name-and-type buffer, and an index buffer. These three buffer are defined as follows:

```
struct CONSTANT_INDEX              constIdx[32767];
struct NAMEANDTYPE_INDEX           const_nt[400];
struct CONSTANT_Utf8_POOL          constUtf8[32767];
```

There are three kinds of information need to be added to the constant pool: 1) constant string, constant class name; 2) strings for constant method or constant field reference; 3) strings representing constant float numbers. The main function that fill in the constant pool is:

```
AddConstantNames(char *name, int type, char *sig, char *retType, char *className)
```

where the `name` contains the constant string, `type` indicating the type of information to be added, for example, constant class name. If the information is for constant method/field reference, the argument `sig` and `retType` represents the signature of the method and the type of the method/field. In this case, the last argument `className` indicating the class that the method/field belongs to.

The function `AddConstantNames` adds the constant string to the Utf8 buffer and create index in one of other buffers or both buffers according to the `type` of information. Before it adds a constant string, it looks up the Utf8 buffer, if the string already exists then the index buffer is checked to see if there's also the index required If no required index, an index is created. If the constant string is not in the Utf8 buffer, it's added into the Utf8 buffer and the corresponding index is created in one of other buffers or both buffers for name-and-type strings.

Later in target code generation, the three buffers are merged and spooled to the class file.

3.6.2 Add Constant Information for System-Defined Method Calls

At present, the compiler can handle a limitted number of system-defined method calls and fiield references. But it can be easily expanded to handle more.

The system-defined methods, fields and classes are kept in the buffer `systemMethod`, `systemField` and `systemClass` respectively. When a system-defined method is called, the compiler looks up these three buffers to resolve the method name and adds the necessary information of the method to the constant pool. This process is called *registering method calls*. It's implemented by the function void `RegisterMethodCall(void)`.

For example, when generating code for the following method call:

```
g.setFont(f);
```

The `RegisterMethodCall` generates:

```
gen("invokevirtual # <Method java.awt.Graphics.setFont(Ljava/awt/Font;)V>",3 -1);
AddConstantNames("java/awt/Graphics", CONSTANT_Class, NULL, NULL, NULL);
AddConstantNames("setFont", CONSTANT_Methodref, "Ljava/awt/Font;", "V",
                 "java/awt/Graphics");
```

3.7 The Bytecode -- The Target Code Generation

The bytecode generation is done after the assembler code is generated. The compiler reads the assembler code, parses the code and translate the code into binaries.

This bytecode is generated in accordance with Java class file format of major version 45, minor version 2.

There are four major functions for generating byte code:

```
void ParseInstruction(char *ins, int *reg, int *value, char *constString, int *type)
int TranslateAssemblyCode(FILE *fin)
int GenClassCode(FILE *fin)
void GenMethodBytecode(FILE *fp)
```

The first three functions process the assmbler code and translate into bytecode. The bytecode, at this point, is stored in the buffer `mClass`. Its definition is as follows:

```
struct METHOD_CLASS_CODE {
        unsigned short code;
        unsigned char  memo;
};
```

where the field `code` is the real bytecode, the `memo` indicates the attribute of `code`. The type of `memo` is defined as follows:

```
#define MC_MethodCount     1
#define MC_MethodNumber    2
#define MC_AccessFlag      3
#define MC_NameIndex       4
#define MC_SignatureIndex  5
#define MC_AttributeCount  6
#define MC_AttributeNumber 7
#define MC_AttributeName   8
#define MC_AttributeLength 9
#define MC_CodeLength      10
#define MC_Attribute       11
#define MC_MaxStacks       12
#define MC_MaxLocals       13
#define MC_Instruction     14
#define MC_Operand         15
#define MC_ExceptionTbLen  16
#define MC_AttributesCount 17
```

Function `GenClassCode` writes header for a method then calls the function `TranslateAssemblyCode` which scans the assembler code and calls the function `ParseInstruction`. `ParseInstruction` parses an instruction line and returns useful information in the arguments. The first argument `ins` holds the name of the instruction. The second argument `reg` holds the register number if any. The third argument `value` holds the value of the operand, if there is an operand. If there is a constant string in the instruction line, the string can be referenced by `constString`. At last, argument `type` is used to distinguish whether the `constString` represents a string or a floating-point number so that a floating-point number can be converted from the IEEE-754 representation.

Function `TranslateAssemblyCode` then use the above returned information to lookup the instruction array to obtain the instruction number, resolve the floating-point number, and generate bytecode to the buffer `mClass`.

Finally, `GenMethodByteCode` is called to spooled the contents in `mClass` to the class file. Other functions called at the last stage are:

```
void WriteFileHeader(FILE *fp, unsigned short minorVersion,
                           unsigned short majorVersion);
void AssembleConstantPool(FILE *fp);
void GenClassHeader(FILE *fp);
```

where fp points to the class file.

3.8 Register Allocation

Register allocation occurs when the parser sees the variable declarations. Field references do not need registers, they use the constant pool. Therefore, register is allocated for only local variables. For a non-static method, the register that can be allocated starts from 1, while for a static method, the register number starts from 0. For more information, please refer to *2.5 Register Allocation*.

The global variable `local_counter` is used to represent the max register number that has been assigned. It's initialized to 0 or 1 based on if current method is a static one. In the parsing process, whenever the parser sees a new variable declared, the `local_counter` is increased by 1 and its value is assigned to the variable as the register number. Therefore, each variable has a different register number in a method. In a static method, register number starts with 0, so the first declared variable has register 0. Also, when the class `this` is referenced, it also uses register 0 in a non-static method.

3.9 Optimization Implementation

3.9.1 Constant Expression (part of constant expression) Evaluation at Compile Time

Information about constants is recorded in the nodes of the parse tree. For detailed structure about the parse tree, please refer to ***3.3 Data Structures***.

When the parser seees an constant (floating point or integer for this compiler), the value of the constant is recorded in the field `const_val` of a node of the parse tree. When an expression or part of the expression is reduced, the compiler checks the possible nodes to see whether the expression so far could be evaluated, for example, take the following rule:

```
AdditiveExpression ->AdditiveExpression '+' MultiplicativeExpression
```

The compiler checks the two nodes beside the plus symbol to see if both nodes have constant values. If they do, the compiler will plus them together and assign the result to the left most node in the rule.

3.9.2 Dead-Code Elimination

This optimization is done within the basic blocks based on live variable analysis. For the algorithm for partitioning the source code into basic blocks, please refer to the ***Compilers -- Principles, Techniques, and Tools*** (by Alfred V. Aho, Ravi Sethi,, and Jeffrey D. Ullman).

The optimization occurs after the code for a method has been generated and before the code is spooled to a temporary file. In this compiler, the function `OptBuf`() which is called by the function `PrintBuf`() implements the optimization. The `goto` addresses (line numbers in this project) is updated both before and after the optimization.

During the optimization, the code is kept in the buffer `buf`, the buffer `buf` is an array of records whose structure is shown as below. Each record contains exactly one instruction line.

```
struct insBuffer {
        int byteCounter;          /* address/line number       */
        char instruction[128];    /* instruction               */
        int offset;               /* length of the instruction */
        int ref;                  /* reference times/lives of the statement */
        int mark;                 /* if the instruction statement is kept*/
        int eos;                  /* end of statement flag */
        int leader;               /* statement leader flag */
};
```

The smallest optimization unit is the statement in the source code or a set of instructions which constitute a statement in the assembler code, which means a statement is either kept or deleted during the optimization. That means, when a variable is assigned a value and never used, it's deleted.

In the parsing process, when a variable is assigned a value, it's reference flag (`ref`) is set to 0. In the following statements before another value is assigned to the same variable, the reference flag is increased by 1 whenever it's referenced. This is called '*adding lives*' to the variables.

In the optimization process, the compiler examins every statement within a basic block in reverse order. First it searches the instruction which constains a `store` operation and the corresponding variable has no lives. Then the compiler examins the right-hand part of the assignment expression. when the optimizer sees a `load (iload/fload)` instruction, it searches the previous closest `store (istotr/fstore)` statement for the same variable and decreases its lives by one because the `load` operation is in a useless assignment statement and the life added by this reference is considered invalid. At last, the compiler marks those useless statement (assignment statements with no lives) as 'killed ' and deleted them finally.

As to the method calls, there is no real instructions for storing values into the variables in the argument list. In this case, the compiler generates psedu-assignment code for those variables. For example, for the method

```
static int  fact(int i) {...},
```

The first instruction will be `istore_0`, but it's marked as 'killed' when it's generated so that it can be deleted at the end of optimization. In this way, when `i` is referenced, its 'lives' can be added.

4. Example Programs and Codes

4.1 Example 1: `test1.java`

Source Code:

```
public class test1 {
```

```
  public static void main(String argv[]) {
    int a = 56; int c= 89; int d =90;
    int b;
         b = 8 * 9 + 10 -7 * (5+8);

        a = a * b + c * (a+b);
        c = a * b - c * d +a;
        d = c + a;
        System.out.println("a is = " + a + "b is = "+ b + "c is " +c);

  }
}
```

Intermefiate Assembler Code:

```
Compiled from test1.java
modifiers class test1 extends java.lang.Object {



        modifiers returnType main ();
        <init>()V;


Method main
        0 bipush 56
        2 istore_1
        4 bipush 89
        6 istore_2
        8 bipush 90
        10 istore_3
        12 bipush 9
        14 ineg
        15 istore_4
        17 iload_1
        19 iload_4
        21 imul
        22 iload_2
        24 iload_1
        26 iload_4
        28 iadd
        29 imul
        30 iadd
        31 istore_1
        33 iload_1
        35 iload_4
        37 imul
        38 iload_2
        40 iload_3
        42 imul
        43 isub
        44 iload_1
        46 iadd
        47 istore_2
        49 getstatic # <Field java.lang.System.out Ljava/io/PrintStream;>
        52 new # <Class java.lang.StringBuffer>
        55 dup
        56 invokenonvirtual # <Method java.lang.StringBuffer.<init>()V>
```

```
        59 ldc # <String "a is = ">
        61 invokevirtual # <Method java.lang.StringBuffer.append(Ljava/lang/
String;)Ljava/lang/StringBuffer;>
        64 iload_1
        66 invokevirtual # <Method java.lang.StringBuffer.append(I)Ljava/lang/
StringBuffer;> ==>0
        69 ldc # <String "b is = ">
        71 invokevirtual # <Method java.lang.StringBuffer.append(Ljava/lang/
String;)Ljava/lang/StringBuffer;>
        74 iload_4
        76 invokevirtual # <Method java.lang.StringBuffer.append(I)Ljava/lang/
StringBuffer;> ==>0
        79 ldc # <String "c is ">
        81 invokevirtual # <Method java.lang.StringBuffer.append(Ljava/lang/
String;)Ljava/lang/StringBuffer;>
        84 iload_2
        86 invokevirtual # <Method java.lang.StringBuffer.append(I)Ljava/lang/
StringBuffer;> ==>0
        89 invokevirtual # <Method java.lang.StringBuffer.toString()Ljava/lang/
String;>
        92 invokevirtual # <Method java.io.PrintStream.println(Ljava/lang/String;)V>
        95 return

Method <init>
        0 aload_0
        2 invokenonvirtual # <Method java.lang.Object.<init>()V>
        5 return
}
```

Interpreted Bytecode:

```
----------------------------------------------------------------------
**********magic**********
ca fe ba be
----------------------------------------------------------------------
**********minor_version**********
0 2
----------------------------------------------------------------------
**********major_version**********
0 45
----------------------------------------------------------------------
**********Constant_Pool_Count**********
0 52
##########Constant_Pool_Start##########
----------------------------------------------------------------------
**********Constant_Pool[1]=**********
8
##########String##########
**********Stringindex**********
0 30
----------------------------------------------------------------------
**********Constant_Pool[2]=**********
8
##########String##########
**********Stringindex**********
0 34
----------------------------------------------------------------------
```

```
**********Constant_Pool[3]=**********
8
##########String##########
**********Stringindex**********
0 35
----------------------------------------------------------------------
**********Constant_Pool[4]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 24
----------------------------------------------------------------------
**********Constant_Pool[5]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 27
----------------------------------------------------------------------
**********Constant_Pool[6]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 38
----------------------------------------------------------------------
**********Constant_Pool[7]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 41
----------------------------------------------------------------------
**********Constant_Pool[8]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 49
----------------------------------------------------------------------
**********Constant_Pool[9]=**********
9
##########Method##########
**********Classindex**********
0 4
**********Name&TypeIndex**********
0 17
----------------------------------------------------------------------
**********Constant_Pool[10]=**********
10
##########MethodRef##########
**********Classindex**********
0 5
```

```
**********Name&TypeIndex**********
0 18
--------------------------------------------------------------------
**********Constant_Pool[11]=**********
10
##########MethodRef##########
**********Classindex**********
0 5
**********Name&TypeIndex**********
0 19
--------------------------------------------------------------------
**********Constant_Pool[12]=**********
10
##########MethodRef##########
**********Classindex**********
0 5
**********Name&TypeIndex**********
0 20
--------------------------------------------------------------------
**********Constant_Pool[13]=**********
10
##########MethodRef##########
**********Classindex**********
0 5
**********Name&TypeIndex**********
0 21
--------------------------------------------------------------------
**********Constant_Pool[14]=**********
10
##########MethodRef##########
**********Classindex**********
0 6
**********Name&TypeIndex**********
0 22
--------------------------------------------------------------------
**********Constant_Pool[15]=**********
10
##########MethodRef##########
**********Classindex**********
0 7
**********Name&TypeIndex**********
0 18
--------------------------------------------------------------------
**********Constant_Pool[16]=**********
10
##########MethodRef##########
**********Classindex**********
0 8
**********Name&TypeIndex**********
0 23
--------------------------------------------------------------------
**********Constant_Pool[17]=**********
12
##########name&Type##########
```

```
**********Nameindex**********
0 25
**********SignatureIndex**********
0 26
---------------------------------------------------------------------
**********Constant_Pool[18]=**********
12
##########name&Type##########
**********Nameindex**********
0 28
**********SignatureIndex**********
0 29
---------------------------------------------------------------------
**********Constant_Pool[19]=**********
12
##########name&Type##########
**********Nameindex**********
0 31
**********SignatureIndex**********
0 32
---------------------------------------------------------------------
**********Constant_Pool[20]=**********
12
##########name&Type##########
**********Nameindex**********
0 31
**********SignatureIndex**********
0 33
---------------------------------------------------------------------
**********Constant_Pool[21]=**********
12
##########name&Type##########
**********Nameindex**********
0 36
**********SignatureIndex**********
0 37
---------------------------------------------------------------------
**********Constant_Pool[22]=**********
12
##########name&Type##########
**********Nameindex**********
0 39
**********SignatureIndex**********
0 40
---------------------------------------------------------------------
**********Constant_Pool[23]=**********
12
##########name&Type##########
**********Nameindex**********
0 50
**********SignatureIndex**********
0 51
---------------------------------------------------------------------
**********Constant_Pool[24]=**********
```

```
1
##########Constant Utf8##########
**********length**********
0 16
Length is 16
##########Actual String##########
java/lang/System
--------------------------------------------------------------------
**********Constant_Pool[25]=**********
1
##########Constant Utf8##########
**********length**********
0 3
Length is 3
##########Actual String##########
out
--------------------------------------------------------------------
**********Constant_Pool[26]=**********
1
##########Constant Utf8##########
**********length**********
0 21
Length is 21
##########Actual String##########
Ljava/io/PrintStream;
--------------------------------------------------------------------
**********Constant_Pool[27]=**********
1
##########Constant Utf8##########
**********length**********
0 22
Length is 22
##########Actual String##########
java/lang/StringBuffer
--------------------------------------------------------------------
**********Constant_Pool[28]=**********
1
##########Constant Utf8##########
**********length**********
0 6
Length is 6
##########Actual String##########
<init>
--------------------------------------------------------------------
**********Constant_Pool[29]=**********
1
##########Constant Utf8##########
**********length**********
0 3
Length is 3
##########Actual String##########
()V
--------------------------------------------------------------------
**********Constant_Pool[30]=**********
```

```
1
##########Constant Utf8##########
**********length**********
0 7
Length is 7
##########Actual String##########
a is =
--------------------------------------------------------------------
**********Constant_Pool[31]=**********
1
##########Constant Utf8##########
**********length**********
0 6
Length is 6
##########Actual String##########
append
--------------------------------------------------------------------
**********Constant_Pool[32]=**********
1
##########Constant Utf8##########
**********length**********
0 44
Length is 44
##########Actual String##########
(Ljava/lang/String;)Ljava/lang/StringBuffer;
--------------------------------------------------------------------
**********Constant_Pool[33]=**********
1
##########Constant Utf8##########
**********length**********
0 27
Length is 27
##########Actual String##########
(I)Ljava/lang/StringBuffer;
--------------------------------------------------------------------
**********Constant_Pool[34]=**********
1
##########Constant Utf8##########
**********length**********
0 7
Length is 7
##########Actual String##########
b is =
--------------------------------------------------------------------
**********Constant_Pool[35]=**********
1
##########Constant Utf8##########
**********length**********
0 5
Length is 5
##########Actual String##########
c is
--------------------------------------------------------------------
**********Constant_Pool[36]=**********
```

```
1
##########Constant Utf8##########
**********length**********
0 8
Length is 8
##########Actual String##########
toString
--------------------------------------------------------------------
**********Constant_Pool[37]=**********
1
##########Constant Utf8##########
**********length**********
0 20
Length is 20
##########Actual String##########
()Ljava/lang/String;
--------------------------------------------------------------------
**********Constant_Pool[38]=**********
1
##########Constant Utf8##########
**********length**********
0 19
Length is 19
##########Actual String##########
java/io/PrintStream
--------------------------------------------------------------------
**********Constant_Pool[39]=**********
1
##########Constant Utf8##########
**********length**********
0 7
Length is 7
##########Actual String##########
println
--------------------------------------------------------------------
**********Constant_Pool[40]=**********
1
##########Constant Utf8##########
**********length**********
0 21
Length is 21
##########Actual String##########
(Ljava/lang/String;)V
--------------------------------------------------------------------
**********Constant_Pool[41]=**********
1
##########Constant Utf8##########
**********length**********
0 16
Length is 16
##########Actual String##########
java/lang/Object
--------------------------------------------------------------------
**********Constant_Pool[42]=**********
```

```
1
##########Constant Utf8##########
**********length**********
0 10
Length is 10
##########Actual String##########
test1.java
--------------------------------------------------------------------
**********Constant_Pool[43]=**********
1
##########Constant Utf8##########
**********length**********
0 13
Length is 13
##########Actual String##########
ConstantValue
--------------------------------------------------------------------
**********Constant_Pool[44]=**********
1
##########Constant Utf8##########
**********length**********
0 10
Length is 10
##########Actual String##########
Exceptions
--------------------------------------------------------------------
**********Constant_Pool[45]=**********
1
##########Constant Utf8##########
**********length**********
0 15
Length is 15
##########Actual String##########
LineNumberTable
--------------------------------------------------------------------
**********Constant_Pool[46]=**********
1
##########Constant Utf8##########
**********length**********
0 10
Length is 10
##########Actual String##########
SourceFile
--------------------------------------------------------------------
**********Constant_Pool[47]=**********
1
##########Constant Utf8##########
**********length**********
0 14
Length is 14
##########Actual String##########
LocalVariables
--------------------------------------------------------------------
**********Constant_Pool[48]=**********
```

```
1
##########Constant Utf8##########
**********length**********
0 4
Length is 4
##########Actual String##########
Code
--------------------------------------------------------------------
**********Constant_Pool[49]=**********
1
##########Constant Utf8##########
**********length**********
0 5
Length is 5
##########Actual String##########
test1
--------------------------------------------------------------------
**********Constant_Pool[50]=**********
1
##########Constant Utf8##########
**********length**********
0 4
Length is 4
##########Actual String##########
main
--------------------------------------------------------------------
**********Constant_Pool[51]=**********
1
##########Constant Utf8##########
**********length**********
0 22
Length is 22
##########Actual String##########
([Ljava/lang/String;)V
--------------------------------------------------------------------
**********access_flags**********
0 1
--------------------------------------------------------------------
**********this_class**********
0 8
--------------------------------------------------------------------
**********super_class**********
0 7
--------------------------------------------------------------------
**********interfaces_count**********
0 0
numberof interfaces 0
--------------------------------------------------------------------
--------------------------------------------------------------------
**********fields_count**********
0 0
numberof fieldscount 0
--------------------------------------------------------------------
--------------------------------------------------------------------
```

```
**********methods_count**********
0 2
numberof fieldscount 2
--------------------------------------------------------------------
&&&&&&&&&&Reading Methosds&&&&&&&&&&
--------------------------------------------------------------------
 Method no: 0
**********Access Flags**********
0 9
**********Name Index **********
0 50
**********Signature Index**********
0 51
**********AttributeCount**********
0 1
Attribute Count is1
--------------------------------------------------------------------
&&&&&&&&&&Reading Attributes&&&&&&&&&&
--------------------------------------------------------------------
Attribute no: 0
**********Attribute Name**********
0 48
**********Attribute Length **********
104
--------------------------------------------------------------------
&&&&&&&&&&Attribbutes&&&&&&&&&&
200
200
0
96
16
56
54
1
16
89
54
2
16
90
54
3
16
9
116
54
4
21
1
21
4
104
21
2
```

21
1
21
4
96
104
96
54
1
21
1
21
4
104
21
2
21
3
104
100
21
1
96
54
2
178
0
9
187
0
5
89
183
0
10
18
1
182
0
11
21
1
182
0
12
18
2
182
0
11
21
4
182
0

```
12
18
3
182
0
11
21
2
182
0
12
182
0
13
182
0
14
177
0
0
0
0
--------------------------------------------------------------------
--------------------------------------------------------------------
 Method no: 1
**********Access Flags**********
0 1
**********Name Index **********
0 28
**********Signature Index**********
0 29
**********AttributeCount**********
0 1
Attribute Count is1
--------------------------------------------------------------------
&&&&&&&&&&Reading Attributes&&&&&&&&&&
--------------------------------------------------------------------
Attribute no: 0
**********Attribute Name**********
0 48
**********Attribute Length **********
14
--------------------------------------------------------------------
&&&&&&&&&&Attribbutes&&&&&&&&&&
200
200
0
6
25
0
183
0
15
177
```

```
0
0
0
0
```


4.2 Example 2: test2.java

Source Code:

```
public class test2 {

  public static void main(String argv[]) {
    int a = 56; int c= 89; int d =90;
    int b;

    float fa,fb;
         b = 8 * 9 + 10 -7 * (5+8);

        a = a * b + c * (a+b);
        c = a * b - c * d +a;
        d = c + a;
        System.out.println("a is = " + a + " b is = "+ b + " c is " +c);
        fa =(float) 78.8*(float)89+ (float)a;
        fb = (float)78.8*(float)89+ (float)b;
        System.out.println(" fa is = " + fa + " fb is = "+ fb );
}
}
```

Intermediate Assembler Code:

```
Compiled from test2.java
modifiers class test2 extends java.lang.Object {

            modifiers returnType main ();
            <init>()V;

Method main
            0 bipush 56
            2 istore_1
            4 bipush 89
            6 istore_2
            8 bipush 90
            10 istore_3
            12 bipush 9
            14 ineg
            15 istore_4
            17 iload_1
            19 iload_4
            21 imul
            22 iload_2
            24 iload_1
            26 iload_4
```

```
            28 iadd
            29 imul
            30 iadd
            31 istore_1
            33 iload_1
            35 iload_4
            37 imul
            38 iload_2
            40 iload_3
            42 imul
            43 isub
            44 iload_1
            46 iadd
            47 istore_2
            49 getstatic # <Field java.lang.System.out Ljava/io/PrintStream;>
            52 new # <Class java.lang.StringBuffer>
            55 dup
            56 invokenonvirtual # <Method java.lang.StringBuffer.<init>()V>
            59 ldc # <String "a is = ">
            61 invokevirtual # <Method java.lang.StringBuffer.append(Ljava/
lang/String;)Ljava/lang/StringBuffer;>
            64 iload_1
            66 invokevirtual # <Method java.lang.StringBuffer.append(I)Ljava/
lang/StringBuffer;> ==>0
            69 ldc # <String " b is = ">
            71 invokevirtual # <Method java.lang.StringBuffer.append(Ljava/
lang/String;)Ljava/lang/StringBuffer;>
            74 iload_4
            76 invokevirtual # <Method java.lang.StringBuffer.append(I)Ljava/
lang/StringBuffer;> ==>0
            79 ldc # <String " c is ">
            81 invokevirtual # <Method java.lang.StringBuffer.append(Ljava/
lang/String;)Ljava/lang/StringBuffer;>
            84 iload_2
            86 invokevirtual # <Method java.lang.StringBuffer.append(I)Ljava/
lang/StringBuffer;> ==>0
            89 invokevirtual # <Method java.lang.StringBuffer.toString()Ljava/
lang/String;>
            92 invokevirtual # <Method java.io.PrintStream.println(Ljava/lang/
String;)V>
            95 ldc # <Real 78.800003           >
            97 bipush 89
            99 i2f
            100 fmul
            101 iload_1
            103 i2f
            104 fadd
            105 fstore_5
            107 ldc # <Real 78.800003           >
            109 bipush 89
            111 i2f
            112 fmul
            113 iload_4
            115 i2f
```

```
            116 fadd
            117 fstore_6
            119 getstatic # <Field java.lang.System.out Ljava/io/PrintStream;>
            122 new # <Class java.lang.StringBuffer>
            125 dup
            126 invokenonvirtual # <Method java.lang.StringBuffer.<init>()V>
            129 ldc # <String " fa is = ">
            131 invokevirtual # <Method java.lang.StringBuffer.append(Ljava/
lang/String;)Ljava/lang/StringBuffer;>
            134 fload_5
            136 invokevirtual # <Method java.lang.StringBuffer.append(F)Ljava/
lang/StringBuffer;> ==>2
            139 ldc # <String " fb is = ">
            141 invokevirtual # <Method java.lang.StringBuffer.append(Ljava/
lang/String;)Ljava/lang/StringBuffer;>
            144 fload_6
            146 invokevirtual # <Method java.lang.StringBuffer.append(F)Ljava/
lang/StringBuffer;> ==>2
            149 invokevirtual # <Method
java.lang.StringBuffer.toString()Ljava/lang/String;>
            152 invokevirtual # <Method java.io.PrintStream.println(Ljava/
lang/String;)V>
            155 return

Method <init>
            0 aload_0
            2 invokenonvirtual # <Method java.lang.Object.<init>()V>
            5 return
}
```

Interpreted Bytecode:

```
-------------------------------------------------------------------
**********magic**********
ca fe ba be
-------------------------------------------------------------------
**********minor_version**********
0 2
-------------------------------------------------------------------
**********major_version**********
0 45
-------------------------------------------------------------------
**********Constant_Pool_Count**********
0 60
##########Constant_Pool_Start##########
-------------------------------------------------------------------
**********Constant_Pool[1]=**********
8
##########String##########
**********Stringindex**********
0 35
-------------------------------------------------------------------
**********Constant_Pool[2]=**********
8
##########String##########
**********Stringindex**********
```

```
0 39
--------------------------------------------------------------------
**********Constant_Pool[3]=**********
8
##########String##########
**********Stringindex**********
0 40
--------------------------------------------------------------------
**********Constant_Pool[4]=**********
8
##########String##########
**********Stringindex**********
0 46
--------------------------------------------------------------------
**********Constant_Pool[5]=**********
8
##########String##########
**********Stringindex**********
0 48
--------------------------------------------------------------------
**********Constant_Pool[6]=**********
4
##########FloatingConstant##########
66 157 153 154
--------------------------------------------------------------------
**********Constant_Pool[7]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 29
--------------------------------------------------------------------
**********Constant_Pool[8]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 32
--------------------------------------------------------------------
**********Constant_Pool[9]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 43
--------------------------------------------------------------------
**********Constant_Pool[10]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 49
--------------------------------------------------------------------
**********Constant_Pool[11]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 57
```

```
-----------------------------------------------------------------------
**********Constant_Pool[12]=**********
9
##########Method##########
**********Classindex**********
0 7
**********Name&TypeIndex**********
0 21
-----------------------------------------------------------------------
**********Constant_Pool[13]=**********
10
##########MethodRef##########
**********Classindex**********
0 8
**********Name&TypeIndex**********
0 22
-----------------------------------------------------------------------
**********Constant_Pool[14]=**********
10
##########MethodRef##########
**********Classindex**********
0 8
**********Name&TypeIndex**********
0 23
-----------------------------------------------------------------------
**********Constant_Pool[15]=**********
10
##########MethodRef##########
**********Classindex**********
0 8
**********Name&TypeIndex**********
0 24
-----------------------------------------------------------------------
**********Constant_Pool[16]=**********
10
##########MethodRef##########
**********Classindex**********
0 8
**********Name&TypeIndex**********
0 25
-----------------------------------------------------------------------
**********Constant_Pool[17]=**********
10
##########MethodRef##########
**********Classindex**********
0 9
**********Name&TypeIndex**********
0 26
-----------------------------------------------------------------------
**********Constant_Pool[18]=**********
10
##########MethodRef##########
**********Classindex**********
0 8
**********Name&TypeIndex**********
0 27
-----------------------------------------------------------------------
**********Constant_Pool[19]=**********
10
```

```
##########MethodRef##########
**********Classindex**********
0 10
**********Name&TypeIndex**********
0 22
--------------------------------------------------------------------
**********Constant_Pool[20]=**********
10
##########MethodRef##########
**********Classindex**********
0 11
**********Name&TypeIndex**********
0 28
--------------------------------------------------------------------
**********Constant_Pool[21]=**********
12
##########name&Type##########
**********Nameindex**********
0 30
**********SignatureIndex**********
0 31
--------------------------------------------------------------------
**********Constant_Pool[22]=**********
12
##########name&Type##########
**********Nameindex**********
0 33
**********SignatureIndex**********
0 34
--------------------------------------------------------------------
**********Constant_Pool[23]=**********
12
##########name&Type##########
**********Nameindex**********
0 36
**********SignatureIndex**********
0 37
--------------------------------------------------------------------
**********Constant_Pool[24]=**********
12
##########name&Type##########
**********Nameindex**********
0 36
**********SignatureIndex**********
0 38
--------------------------------------------------------------------
**********Constant_Pool[25]=**********
12
##########name&Type##########
**********Nameindex**********
0 41
**********SignatureIndex**********
0 42
--------------------------------------------------------------------
**********Constant_Pool[26]=**********
12
##########name&Type##########
**********Nameindex**********
0 44
```

```
**********SignatureIndex**********
0 45
--------------------------------------------------------------------
**********Constant_Pool[27]=**********
12
##########name&Type##########
**********Nameindex**********
0 36
**********SignatureIndex**********
0 47
--------------------------------------------------------------------
**********Constant_Pool[28]=**********
12
##########name&Type##########
**********Nameindex**********
0 58
**********SignatureIndex**********
0 59
--------------------------------------------------------------------
**********Constant_Pool[29]=**********
1
##########Constant Utf8##########
**********length**********
0 16
Length is 16
##########Actual String##########
java/lang/System
--------------------------------------------------------------------
**********Constant_Pool[30]=**********
1
##########Constant Utf8##########
**********length**********
0 3
Length is 3
##########Actual String##########
out
--------------------------------------------------------------------
**********Constant_Pool[31]=**********
1
##########Constant Utf8##########
**********length**********
0 21
Length is 21
##########Actual String##########
Ljava/io/PrintStream;
--------------------------------------------------------------------
**********Constant_Pool[32]=**********
1
##########Constant Utf8##########
**********length**********
0 22
Length is 22
##########Actual String##########
java/lang/StringBuffer
--------------------------------------------------------------------
**********Constant_Pool[33]=**********
1
##########Constant Utf8##########
**********length**********
```

```
0 6
Length is 6
##########Actual String##########
<init>
--------------------------------------------------------------------
**********Constant_Pool[34]=**********
1
##########Constant Utf8##########
**********length**********
0 3
Length is 3
##########Actual String##########
()V
--------------------------------------------------------------------
**********Constant_Pool[35]=**********
1
##########Constant Utf8##########
**********length**********
0 7
Length is 7
##########Actual String##########
a is =
--------------------------------------------------------------------
**********Constant_Pool[36]=**********
1
##########Constant Utf8##########
**********length**********
0 6
Length is 6
##########Actual String##########
append
--------------------------------------------------------------------
**********Constant_Pool[37]=**********
1
##########Constant Utf8##########
**********length**********
0 44
Length is 44
##########Actual String##########
(Ljava/lang/String;)Ljava/lang/StringBuffer;
--------------------------------------------------------------------
**********Constant_Pool[38]=**********
1
##########Constant Utf8##########
**********length**********
0 27
Length is 27
##########Actual String##########
(I)Ljava/lang/StringBuffer;
--------------------------------------------------------------------
**********Constant_Pool[39]=**********
1
##########Constant Utf8##########
**********length**********
0 8
Length is 8
##########Actual String##########
 b is =
--------------------------------------------------------------------
```

```
**********Constant_Pool[40]=**********
1
##########Constant Utf8##########
**********length**********
0 6
Length is 6
##########Actual String##########
 c is
---------------------------------------------------------------------
**********Constant_Pool[41]=**********
1
##########Constant Utf8##########
**********length**********
0 8
Length is 8
##########Actual String##########
toString
---------------------------------------------------------------------
**********Constant_Pool[42]=**********
1
##########Constant Utf8##########
**********length**********
0 20
Length is 20
##########Actual String##########
()Ljava/lang/String;
---------------------------------------------------------------------
**********Constant_Pool[43]=**********
1
##########Constant Utf8##########
**********length**********
0 19
Length is 19
##########Actual String##########
java/io/PrintStream
---------------------------------------------------------------------
**********Constant_Pool[44]=**********
1
##########Constant Utf8##########
**********length**********
0 7
Length is 7
##########Actual String##########
println
---------------------------------------------------------------------
**********Constant_Pool[45]=**********
1
##########Constant Utf8##########
**********length**********
0 21
Length is 21
##########Actual String##########
(Ljava/lang/String;)V
---------------------------------------------------------------------
**********Constant_Pool[46]=**********
1
##########Constant Utf8##########
**********length**********
0 9
```

```
Length is 9
##########Actual String##########
 fa is =
---------------------------------------------------------------------
**********Constant_Pool[47]=**********
1
##########Constant Utf8##########
**********length**********
0 27
Length is 27
##########Actual String##########
(F)Ljava/lang/StringBuffer;
---------------------------------------------------------------------
**********Constant_Pool[48]=**********
1
##########Constant Utf8##########
**********length**********
0 9
Length is 9
##########Actual String##########
 fb is =
---------------------------------------------------------------------
**********Constant_Pool[49]=**********
1
##########Constant Utf8##########
**********length**********
0 16
Length is 16
##########Actual String##########
java/lang/Object
---------------------------------------------------------------------
**********Constant_Pool[50]=**********
1
##########Constant Utf8##########
**********length**********
0 10
Length is 10
##########Actual String##########
test2.java
---------------------------------------------------------------------
**********Constant_Pool[51]=**********
1
##########Constant Utf8##########
**********length**********
0 13
Length is 13
##########Actual String##########
ConstantValue
---------------------------------------------------------------------
**********Constant_Pool[52]=**********
1
##########Constant Utf8##########
**********length**********
0 10
Length is 10
##########Actual String##########
Exceptions
---------------------------------------------------------------------
**********Constant_Pool[53]=**********
```

```
1
##########Constant Utf8##########
**********length**********
0 15
Length is 15
##########Actual String##########
LineNumberTable
---------------------------------------------------------------------
**********Constant_Pool[54]=**********
1
##########Constant Utf8##########
**********length**********
0 10
Length is 10
##########Actual String##########
SourceFile
---------------------------------------------------------------------
**********Constant_Pool[55]=**********
1
##########Constant Utf8##########
**********length**********
0 14
Length is 14
##########Actual String##########
LocalVariables
---------------------------------------------------------------------
**********Constant_Pool[56]=**********
1
##########Constant Utf8##########
**********length**********
0 4
Length is 4
##########Actual String##########
Code
---------------------------------------------------------------------
**********Constant_Pool[57]=**********
1
##########Constant Utf8##########
**********length**********
0 5
Length is 5
##########Actual String##########
test2
---------------------------------------------------------------------
**********Constant_Pool[58]=**********
1
##########Constant Utf8##########
**********length**********
0 4
Length is 4
##########Actual String##########
main
---------------------------------------------------------------------
**********Constant_Pool[59]=**********
1
##########Constant Utf8##########
**********length**********
0 22
Length is 22
```

```
##########Actual String##########
([Ljava/lang/String;)V
---------------------------------------------------------------------
**********access_flags**********
0 1
---------------------------------------------------------------------
**********this_class**********
0 11
---------------------------------------------------------------------
**********super_class**********
0 10
---------------------------------------------------------------------
**********interfaces_count**********
0 0
numberof interfaces 0
---------------------------------------------------------------------
---------------------------------------------------------------------
**********fields_count**********
0 0
numberof fieldscount 0
---------------------------------------------------------------------
---------------------------------------------------------------------
**********methods_count**********
0 2
numberof fieldscount 2
---------------------------------------------------------------------
&&&&&&&&&&Reading Methosds&&&&&&&&&&
---------------------------------------------------------------------
 Method no: 0
**********Access Flags**********
0 9
**********Name Index **********
0 58
**********Signature Index**********
0 59
**********AttributeCount**********
0 1
Attribute Count is1
---------------------------------------------------------------------
&&&&&&&&&&Reading Attributes&&&&&&&&&&
---------------------------------------------------------------------
Attribute no: 0
**********Attribute Name**********
0 56
**********Attribute Length **********
164
---------------------------------------------------------------------
&&&&&&&&&&Attribbutes&&&&&&&&&&
200
200
0
156
16
56
54
1
16
89
54
```

2
16
90
54
3
16
9
116
54
4
21
1
21
4
104
21
2
21
1
21
4
96
104
96
54
1
21
1
21
4
104
21
2
21
3
104
100
21
1
96
54
2
178
0
12
187
0
8
89
183
0
13
18
1
182
0
14
21
1

182
0
15
18
2
182
0
14
21
4
182
0
15
18
3
182
0
14
21
2
182
0
15
182
0
16
182
0
17
18
6
16
89
134
106
21
1
134
98
56
5
18
6
16
89
134
106
21
4
134
98
56
6
178
0
12
187
0
8

```
89
183
0
13
18
4
182
0
14
23
5
182
0
18
18
5
182
0
14
23
6
182
0
18
182
0
16
182
0
17
177
0
0
0
0
--------------------------------------------------------------------
--------------------------------------------------------------------
 Method no: 1
**********Access Flags**********
0 1
**********Name Index **********
0 33
**********Signature Index**********
0 34
**********AttributeCount**********
0 1
Attribute Count is1
--------------------------------------------------------------------
&&&&&&&&&&Reading Attributes&&&&&&&&&&
--------------------------------------------------------------------
Attribute no: 0
**********Attribute Name**********
0 56
**********Attribute Length **********
14
--------------------------------------------------------------------
&&&&&&&&&&Attribbutes&&&&&&&&&&
200
200
```

```
0
6
25
0
183
0
19
177
0
0
0
0
--------------------------------------------------------------------
--------------------------------------------------------------------
**********attributes_count**********
0 0
numberof attributescount 0
--------------------------------------------------------------------
--------------------------------------------------------------------
```

4.3 Example 3: test3.java

Source Code:

```
public class test3 {

  public static void main(String argv[]) {
    int a = 56; int c= 89; int d =90;
    int b;
     int[] arr = new int[10];
      float[] farr = new float[10];
    float fa,fb;
         b = 8 * 9 + 10 -7 * (5+8);

        a = a * b + c * (a+b);
        c = a * b - c * d +a;
        d = c + a;
        System.out.println("a is = " + a + "b is = "+ b + "c is " +c);
        fa = (float) 78.8*89+ (float)a;
        fb = (float) 78.8*89+ (float)b;
       System.out.println("fa is = " + fa + "fb is = "+ fb);
    { int i;
     for (i=0; i< 10; i++)
      {arr[i] = a+b+i;
       farr[i] = (float) (fa + fb +  4.5*(float) i);
     }
     for (i=0; i< 10; i++)
    System.out.println(farr[i]+" "+arr[i]);
}
}
}
```

Intermediate Assembler Code:

```
Compiled from test3.java
modifiers class test3 extends java.lang.Object {

             modifiers returnType main ();
             <init>()V;
```

```
Method main
            0 bipush 56
            2 istore_1
            4 bipush 89
            6 istore_2
            8 bipush 90
            10 istore_3
            12 bipush 10
            14 newarray 10
            16 astore_5
            18 bipush 10
            20 newarray 6
            22 astore_6
            24 bipush 9
            26 ineg
            27 istore_4
            29 iload_1
            31 iload_4
            33 imul
            34 iload_2
            36 iload_1
            38 iload_4
            40 iadd
            41 imul
            42 iadd
            43 istore_1
            45 iload_1
            47 iload_4
            49 imul
            50 iload_2
            52 iload_3
            54 imul
            55 isub
            56 iload_1
            58 iadd
            59 istore_2
            61 getstatic # <Field java.lang.System.out Ljava/io/PrintStream;>
            64 new # <Class java.lang.StringBuffer>
            67 dup
            68 invokenonvirtual # <Method java.lang.StringBuffer.<init>()V>
            71 ldc # <String "a is = ">
            73 invokevirtual # <Method java.lang.StringBuffer.append(Ljava/lang/
String;)Ljava/lang/StringBuffer;>
            76 iload_1
            78 invokevirtual # <Method java.lang.StringBuffer.append(I)Ljava/lang/
StringBuffer;> ==>0
            81 ldc # <String "b is = ">
            83 invokevirtual # <Method java.lang.StringBuffer.append(Ljava/lang/
String;)Ljava/lang/StringBuffer;>
            86 iload_4
            88 invokevirtual # <Method java.lang.StringBuffer.append(I)Ljava/lang/
StringBuffer;> ==>0
            91 ldc # <String "c is ">
            93 invokevirtual # <Method java.lang.StringBuffer.append(Ljava/lang/
String;)Ljava/lang/StringBuffer;>
            96 iload_2
            98 invokevirtual # <Method java.lang.StringBuffer.append(I)Ljava/lang/
StringBuffer;> ==>0
```

```
            101 invokevirtual # <Method java.lang.StringBuffer.toString()Ljava/lang/
String;>
            104 invokevirtual # <Method java.io.PrintStream.println(Ljava/lang/
String;)V>
            107 ldc # <Real 78.800003           >
            109 bipush 89
            111 i2f
            112 fmul
            113 iload_1
            115 i2f
            116 fadd
            117 fstore_7
            119 ldc # <Real 78.800003           >
            121 bipush 89
            123 i2f
            124 fmul
            125 iload_4
            127 i2f
            128 fadd
            129 fstore_8
            131 getstatic # <Field java.lang.System.out Ljava/io/PrintStream;>
            134 new # <Class java.lang.StringBuffer>
            137 dup
            138 invokenonvirtual # <Method java.lang.StringBuffer.<init>()V>
            141 ldc # <String "fa is = ">
            143 invokevirtual # <Method java.lang.StringBuffer.append(Ljava/lang/
String;)Ljava/lang/StringBuffer;>
            146 fload_7
            148 invokevirtual # <Method java.lang.StringBuffer.append(F)Ljava/lang/
StringBuffer;> ==>2
            151 ldc # <String "fb is = ">
            153 invokevirtual # <Method java.lang.StringBuffer.append(Ljava/lang/
String;)Ljava/lang/StringBuffer;>
            156 fload_8
            158 invokevirtual # <Method java.lang.StringBuffer.append(F)Ljava/lang/
StringBuffer;> ==>2
            161 invokevirtual # <Method java.lang.StringBuffer.toString()Ljava/lang/
String;>
            164 invokevirtual # <Method java.io.PrintStream.println(Ljava/lang/
String;)V>
            167 bipush 0
            169 istore_9
            171 iload_9
            173 bipush 10
            175 if_icmplt 187
            178 goto 220
            181 iinc 9 1
            184 goto 171
            187 aload_5
            189 iload_9
            191 iload_1
            193 iload_4
            195 iadd
            196 iload_9
            198 iadd
            199 iastore
            200 aload_6
            202 iload_9
            204 fload_7
```

```
            206 fload_8
            208 fadd
            209 iload_9
            211 i2f
            212 ldc # <Real 4.500000            >
            214 fmul
            215 fadd
            216 fastore
            217 goto 181
            220 bipush 0
            222 istore_9
            224 iload_9
            226 bipush 10
            228 if_icmplt 240
            231 goto 280
            234 iinc 9 1
            237 goto 224
            240 getstatic # <Field java.lang.System.out Ljava/io/PrintStream;>
            243 new # <Class java.lang.StringBuffer>
            246 dup
            247 invokenonvirtual # <Method java.lang.StringBuffer.<init>()V>
            250 aload_6
            252 iload_9
            254 faload
            255 invokevirtual # <Method java.lang.StringBuffer.append(F)Ljava/lang/
StringBuffer;> ==>2
            258 ldc # <String " ">
            260 invokevirtual # <Method java.lang.StringBuffer.append(Ljava/lang/
String;)Ljava/lang/StringBuffer;>
            263 aload_5
            265 iload_9
            267 iaload
            268 invokevirtual # <Method java.lang.StringBuffer.append(I)Ljava/lang/
StringBuffer;> ==>0
            271 invokevirtual # <Method java.lang.StringBuffer.toString()Ljava/lang/
String;>
            274 invokevirtual # <Method java.io.PrintStream.println(Ljava/lang/
String;)V>
            277 goto 234
            280 return

Method <init>
            0 aload_0
            2 invokenonvirtual # <Method java.lang.Object.<init>()V>
            5 return
}
```

Interpreted Bytecode:

```
--------------------------------------------------------------------
**********magic**********
ca fe ba be
--------------------------------------------------------------------
**********minor_version**********
0 2
--------------------------------------------------------------------
**********major_version**********
0 45
--------------------------------------------------------------------
```

```
**********Constant_Pool_Count**********
0 63
##########Constant_Pool_Start##########
----------------------------------------------------------------------
**********Constant_Pool[1]=**********
8
##########String##########
**********Stringindex**********
0 37
----------------------------------------------------------------------
**********Constant_Pool[2]=**********
8
##########String##########
**********Stringindex**********
0 41
----------------------------------------------------------------------
**********Constant_Pool[3]=**********
8
##########String##########
**********Stringindex**********
0 42
----------------------------------------------------------------------
**********Constant_Pool[4]=**********
8
##########String##########
**********Stringindex**********
0 48
----------------------------------------------------------------------
**********Constant_Pool[5]=**********
8
##########String##########
**********Stringindex**********
0 50
----------------------------------------------------------------------
**********Constant_Pool[6]=**********
8
##########String##########
**********Stringindex**********
0 51
----------------------------------------------------------------------
**********Constant_Pool[7]=**********
4
##########FloatingConstant##########
66 157 153 154
----------------------------------------------------------------------
**********Constant_Pool[8]=**********
4
##########FloatingConstant##########
64 144 0 0
----------------------------------------------------------------------
**********Constant_Pool[9]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 31
----------------------------------------------------------------------
**********Constant_Pool[10]=**********
7
```

```
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 34
--------------------------------------------------------------------
**********Constant_Pool[11]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 45
--------------------------------------------------------------------
**********Constant_Pool[12]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 52
--------------------------------------------------------------------
**********Constant_Pool[13]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 60
--------------------------------------------------------------------
**********Constant_Pool[14]=**********
9
##########Method##########
**********Classindex**********
0 9
**********Name&TypeIndex**********
0 23
--------------------------------------------------------------------
**********Constant_Pool[15]=**********
10
##########MethodRef##########
**********Classindex**********
0 10
**********Name&TypeIndex**********
0 24
--------------------------------------------------------------------
**********Constant_Pool[16]=**********
10
##########MethodRef##########
**********Classindex**********
0 10
**********Name&TypeIndex**********
0 25
--------------------------------------------------------------------
**********Constant_Pool[17]=**********
10
##########MethodRef##########
**********Classindex**********
0 10
**********Name&TypeIndex**********
0 26
--------------------------------------------------------------------
**********Constant_Pool[18]=**********
```

```
10
#########MethodRef##########
**********Classindex**********
0 10
**********Name&TypeIndex**********
0 27
--------------------------------------------------------------------
**********Constant_Pool[19]=**********
10
#########MethodRef##########
**********Classindex**********
0 11
**********Name&TypeIndex**********
0 28
--------------------------------------------------------------------
**********Constant_Pool[20]=**********
10
#########MethodRef##########
**********Classindex**********
0 10
**********Name&TypeIndex**********
0 29
--------------------------------------------------------------------
**********Constant_Pool[21]=**********
10
#########MethodRef##########
**********Classindex**********
0 12
**********Name&TypeIndex**********
0 24
--------------------------------------------------------------------
**********Constant_Pool[22]=**********
10
#########MethodRef##########
**********Classindex**********
0 13
**********Name&TypeIndex**********
0 30
--------------------------------------------------------------------
**********Constant_Pool[23]=**********
12
#########name&Type##########
**********Nameindex**********
0 32
**********SignatureIndex**********
0 33
--------------------------------------------------------------------
**********Constant_Pool[24]=**********
12
#########name&Type##########
**********Nameindex**********
0 35
**********SignatureIndex**********
0 36
--------------------------------------------------------------------
**********Constant_Pool[25]=**********
12
#########name&Type##########
**********Nameindex**********
```

```
0 38
**********SignatureIndex**********
0 39
---------------------------------------------------------------------
**********Constant_Pool[26]=**********
12
##########name&Type##########
**********Nameindex**********
0 38
**********SignatureIndex**********
0 40
---------------------------------------------------------------------
**********Constant_Pool[27]=**********
12
##########name&Type##########
**********Nameindex**********
0 43
**********SignatureIndex**********
0 44
---------------------------------------------------------------------
**********Constant_Pool[28]=**********
12
##########name&Type##########
**********Nameindex**********
0 46
**********SignatureIndex**********
0 47
---------------------------------------------------------------------
**********Constant_Pool[29]=**********
12
##########name&Type##########
**********Nameindex**********
0 38
**********SignatureIndex**********
0 49
---------------------------------------------------------------------
**********Constant_Pool[30]=**********
12
##########name&Type##########
**********Nameindex**********
0 61
**********SignatureIndex**********
0 62
---------------------------------------------------------------------
**********Constant_Pool[31]=**********
1
##########Constant Utf8##########
**********length**********
0 16
Length is 16
##########Actual String##########
java/lang/System
---------------------------------------------------------------------
**********Constant_Pool[32]=**********
1
##########Constant Utf8##########
**********length**********
0 3
Length is 3
```

```
##########Actual String##########
out
---------------------------------------------------------------------
**********Constant_Pool[33]=**********
1
##########Constant Utf8##########
**********length**********
0 21
Length is 21
##########Actual String##########
Ljava/io/PrintStream;
---------------------------------------------------------------------
**********Constant_Pool[34]=**********
1
##########Constant Utf8##########
**********length**********
0 22
Length is 22
##########Actual String##########
java/lang/StringBuffer
---------------------------------------------------------------------
**********Constant_Pool[35]=**********
1
##########Constant Utf8##########
**********length**********
0 6
Length is 6
##########Actual String##########
<init>
---------------------------------------------------------------------
**********Constant_Pool[36]=**********
1
##########Constant Utf8##########
**********length**********
0 3
Length is 3
##########Actual String##########
()V
---------------------------------------------------------------------
**********Constant_Pool[37]=**********
1
##########Constant Utf8##########
**********length**********
0 7
Length is 7
##########Actual String##########
a is =
---------------------------------------------------------------------
**********Constant_Pool[38]=**********
1
##########Constant Utf8##########
**********length**********
0 6
Length is 6
##########Actual String##########
append
---------------------------------------------------------------------
**********Constant_Pool[39]=**********
1
```

```
##########Constant Utf8##########
**********length**********
0 44
Length is 44
##########Actual String##########
(Ljava/lang/String;)Ljava/lang/StringBuffer;
--------------------------------------------------------------------
**********Constant_Pool[40]=**********
1
##########Constant Utf8##########
**********length**********
0 27
Length is 27
##########Actual String##########
(I)Ljava/lang/StringBuffer;
--------------------------------------------------------------------
**********Constant_Pool[41]=**********
1
##########Constant Utf8##########
**********length**********
0 7
Length is 7
##########Actual String##########
b is =
--------------------------------------------------------------------
**********Constant_Pool[42]=**********
1
##########Constant Utf8##########
**********length**********
0 5
Length is 5
##########Actual String##########
c is
--------------------------------------------------------------------
**********Constant_Pool[43]=**********
1
##########Constant Utf8##########
**********length**********
0 8
Length is 8
##########Actual String##########
toString
--------------------------------------------------------------------
**********Constant_Pool[44]=**********
1
##########Constant Utf8##########
**********length**********
0 20
Length is 20
##########Actual String##########
()Ljava/lang/String;
--------------------------------------------------------------------
**********Constant_Pool[45]=**********
1
##########Constant Utf8##########
**********length**********
0 19
Length is 19
##########Actual String##########
```

```
java/io/PrintStream
---------------------------------------------------------------------
**********Constant_Pool[46]=**********
1
##########Constant Utf8##########
**********length**********
0 7
Length is 7
##########Actual String##########
println
---------------------------------------------------------------------
**********Constant_Pool[47]=**********
1
##########Constant Utf8##########
**********length**********
0 21
Length is 21
##########Actual String##########
(Ljava/lang/String;)V
---------------------------------------------------------------------
**********Constant_Pool[48]=**********
1
##########Constant Utf8##########
**********length**********
0 8
Length is 8
##########Actual String##########
fa is =
---------------------------------------------------------------------
**********Constant_Pool[49]=**********
1
##########Constant Utf8##########
**********length**********
0 27
Length is 27
##########Actual String##########
(F)Ljava/lang/StringBuffer;
---------------------------------------------------------------------
**********Constant_Pool[50]=**********
1
##########Constant Utf8##########
**********length**********
0 8
Length is 8
##########Actual String##########
fb is =
---------------------------------------------------------------------
**********Constant_Pool[51]=**********
1
##########Constant Utf8##########
**********length**********
0 1
Length is 1
##########Actual String##########

---------------------------------------------------------------------
**********Constant_Pool[52]=**********
1
##########Constant Utf8##########
```

```
**********length**********
0 16
Length is 16
##########Actual String##########
java/lang/Object
--------------------------------------------------------------------
**********Constant_Pool[53]=**********
1
##########Constant Utf8##########
**********length**********
0 10
Length is 10
##########Actual String##########
test3.java
--------------------------------------------------------------------
**********Constant_Pool[54]=**********
1
##########Constant Utf8##########
**********length**********
0 13
Length is 13
##########Actual String##########
ConstantValue
--------------------------------------------------------------------
**********Constant_Pool[55]=**********
1
##########Constant Utf8##########
**********length**********
0 10
Length is 10
##########Actual String##########
Exceptions
--------------------------------------------------------------------
**********Constant_Pool[56]=**********
1
##########Constant Utf8##########
**********length**********
0 15
Length is 15
##########Actual String##########
LineNumberTable
--------------------------------------------------------------------
**********Constant_Pool[57]=**********
1
##########Constant Utf8##########
**********length**********
0 10
Length is 10
##########Actual String##########
SourceFile
--------------------------------------------------------------------
**********Constant_Pool[58]=**********
1
##########Constant Utf8##########
**********length**********
0 14
Length is 14
##########Actual String##########
LocalVariables
```

```
----------------------------------------------------------------------
**********Constant_Pool[59]=**********
1
##########Constant Utf8##########
**********length**********
0 4
Length is 4
##########Actual String##########
Code
----------------------------------------------------------------------
**********Constant_Pool[60]=**********
1
##########Constant Utf8##########
**********length**********
0 5
Length is 5
##########Actual String##########
test3
----------------------------------------------------------------------
**********Constant_Pool[61]=**********
1
##########Constant Utf8##########
**********length**********
0 4
Length is 4
##########Actual String##########
main
----------------------------------------------------------------------
**********Constant_Pool[62]=**********
1
##########Constant Utf8##########
**********length**********
0 22
Length is 22
##########Actual String##########
([Ljava/lang/String;)V
----------------------------------------------------------------------
**********access_flags**********
0 1
----------------------------------------------------------------------
**********this_class**********
0 13
----------------------------------------------------------------------
**********super_class**********
0 12
----------------------------------------------------------------------
**********interfaces_count**********
0 0
numberof interfaces 0
----------------------------------------------------------------------
----------------------------------------------------------------------
**********fields_count**********
0 0
numberof fieldscount 0
----------------------------------------------------------------------
----------------------------------------------------------------------
**********methods_count**********
0 2
numberof fieldscount 2
```

```
--------------------------------------------------------------------
&&&&&&&&&&Reading Methosds&&&&&&&&&&
--------------------------------------------------------------------
 Method no: 0
**********Access Flags**********
0 9
**********Name Index **********
0 61
**********Signature Index**********
0 62
**********AttributeCount**********
0 1
Attribute Count is1
--------------------------------------------------------------------
&&&&&&&&&&Reading Attributes&&&&&&&&&&
--------------------------------------------------------------------
Attribute no: 0
**********Attribute Name**********
0 59
**********Attribute Length **********
289
--------------------------------------------------------------------
&&&&&&&&&&Attribbutes&&&&&&&&&&
200
200
1
25
16
56
54
1
16
89
54
2
16
90
54
3
16
10
188
10
58
5
16
10
188
6
58
6
16
9
116
54
4
21
1
21
```

4
104
21
2
21
1
21
4
96
104
96
54
1
21
1
21
4
104
21
2
21
3
104
100
21
1
96
54
2
178
0
14
187
0
10
89
183
0
15
18
1
182
0
16
21
1
182
0
17
18
2
182
0
16
21
4
182
0
17

18
3
182
0
16
21
2
182
0
17
182
0
18
182
0
19
18
7
16
89
134
106
21
1
134
98
56
7
18
7
16
89
134
106
21
4
134
98
56
8
178
0
14
187
0
10
89
183
0
15
18
4
182
0
16
23
7
182
0

20
18
5
182
0
16
23
8
182
0
20
182
0
18
182
0
19
16
0
54
9
21
9
16
10
161
0
12
167
0
42
132
9
1
167
255
243
25
5
21
9
21
1
21
4
96
21
9
96
79
25
6
21
9
23
7
23
8
98

21
9
134
18
8
106
98
81
167
255
220
16
0
54
9
21
9
16
10
161
0
12
167
0
49
132
9
1
167
255
243
178
0
14
187
0
10
89
183
0
15
25
6
21
9
48
182
0
20
18
6
182
0
16
25
5
21
9
46

```
182
0
17
182
0
18
182
0
19
167
255
213
177
0
0
0
0
--------------------------------------------------------------------
--------------------------------------------------------------------
 Method no: 1
**********Access Flags**********
0 1
**********Name Index **********
0 35
**********Signature Index**********
0 36
**********AttributeCount**********
0 1
Attribute Count is1
--------------------------------------------------------------------
&&&&&&&&&&Reading Attributes&&&&&&&&&&
--------------------------------------------------------------------
Attribute no: 0
**********Attribute Name**********
0 59
**********Attribute Length **********
14
--------------------------------------------------------------------
&&&&&&&&&&Attribbutes&&&&&&&&&&
200
200
0
6
25
0
183
0
21
177
0
0
0
0
--------------------------------------------------------------------
--------------------------------------------------------------------
**********attributes_count**********
0 0
numberof attributescount 0
--------------------------------------------------------------------
```

4.4 *Example 4:* `test4.java`

Source Code:

```
public class test4 {

  public static void main(String argv[]) {
    int a = 56; int c= 89; int d =90;
    int b;

         b = 8 * 9 + 10 -7 * (5+8);

        a = a * b + c * (a+b);
        c = a * b - c * d +a;
        d = c + a;

      if (b > 89)                          { System.out.println("hello");}
         else { if ((a > 90) || ( d > 89)) { System.out.println("hello1");}
                else                       { System.out.println("hello2");}
              }
      System.out.println("hello3");
  }
}
```

Intermediate Assembler Code:

```
Compiled from test4.java
modifiers class test4 extends java.lang.Object {

             modifiers returnType main ();
             <init>()V;

Method main
             0 bipush 56
             2 istore_1
             4 bipush 89
             6 istore_2
             8 bipush 90
             10 istore_3
             12 bipush 9
             14 ineg
             15 istore_4
             17 iload_1
             19 iload_4
             21 imul
             22 iload_2
             24 iload_1
             26 iload_4
             28 iadd
             29 imul
             30 iadd
             31 istore_1
             33 iload_1
             35 iload_4
             37 imul
             38 iload_2
```

```
            40 iload_3
            42 imul
            43 isub
            44 iload_1
            46 iadd
            47 istore_2
            49 iload_2
            51 iload_1
            53 iadd
            54 istore_3
            56 iload_4
            58 bipush 89
            60 if_icmpgt 66
            63 goto 90
            66 getstatic # <Field java.lang.System.out Ljava/io/PrintStream;>
            69 new # <Class java.lang.StringBuffer>
            72 dup
            73 invokenonvirtual # <Method java.lang.StringBuffer.<init>()V>
            76 ldc # <String "hello">
            78 invokevirtual # <Method java.lang.StringBuffer.append(Ljava/lang/
String;)Ljava/lang/StringBuffer;>
            81 invokevirtual # <Method java.lang.StringBuffer.toString()Ljava/lang/
String;>
            84 invokevirtual # <Method java.io.PrintStream.println(Ljava/lang/
String;)V>
            87 goto 155
            90 iload_1
            92 bipush 90
            94 if_icmpgt 110
            97 goto 100
            100 iload_3
            102 bipush 89
            104 if_icmpgt 110
            107 goto 134
            110 getstatic # <Field java.lang.System.out Ljava/io/PrintStream;>
            113 new # <Class java.lang.StringBuffer>
            116 dup
            117 invokenonvirtual # <Method java.lang.StringBuffer.<init>()V>
            120 ldc # <String "hello1">
            122 invokevirtual # <Method java.lang.StringBuffer.append(Ljava/lang/
String;)Ljava/lang/StringBuffer;>
            125 invokevirtual # <Method java.lang.StringBuffer.toString()Ljava/lang/
String;>
            128 invokevirtual # <Method java.io.PrintStream.println(Ljava/lang/
String;)V>
            131 goto 155
            134 getstatic # <Field java.lang.System.out Ljava/io/PrintStream;>
            137 new # <Class java.lang.StringBuffer>
            140 dup
            141 invokenonvirtual # <Method java.lang.StringBuffer.<init>()V>
            144 ldc # <String "hello2">
            146 invokevirtual # <Method java.lang.StringBuffer.append(Ljava/lang/
String;)Ljava/lang/StringBuffer;>
            149 invokevirtual # <Method java.lang.StringBuffer.toString()Ljava/lang/
String;>
            152 invokevirtual # <Method java.io.PrintStream.println(Ljava/lang/
String;)V>
            155 getstatic # <Field java.lang.System.out Ljava/io/PrintStream;>
            158 new # <Class java.lang.StringBuffer>
```

```
            161 dup
            162 invokenonvirtual # <Method java.lang.StringBuffer.<init>()V>
            165 ldc # <String "hello3">
            167 invokevirtual # <Method java.lang.StringBuffer.append(Ljava/lang/
String;)Ljava/lang/StringBuffer;>
            170 invokevirtual # <Method java.lang.StringBuffer.toString()Ljava/lang/
String;>
            173 invokevirtual # <Method java.io.PrintStream.println(Ljava/lang/
String;)V>
            176 return

Method <init>
            0 aload_0
            2 invokenonvirtual # <Method java.lang.Object.<init>()V>
            5 return
}
```

Interpreted Bytecode:

```
---------------------------------------------------------------------
**********magic**********
ca fe ba be
---------------------------------------------------------------------
**********minor_version**********
0 2
---------------------------------------------------------------------
**********major_version**********
0 45
---------------------------------------------------------------------
**********Constant_Pool_Count**********
0 51
##########Constant_Pool_Start##########
---------------------------------------------------------------------
**********Constant_Pool[1]=**********
8
##########String##########
**********Stringindex**********
0 29
---------------------------------------------------------------------
**********Constant_Pool[2]=**********
8
##########String##########
**********Stringindex**********
0 37
---------------------------------------------------------------------
**********Constant_Pool[3]=**********
8
##########String##########
**********Stringindex**********
0 38
---------------------------------------------------------------------
**********Constant_Pool[4]=**********
8
##########String##########
**********Stringindex**********
0 39
---------------------------------------------------------------------
**********Constant_Pool[5]=**********
7
```

```
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 23
--------------------------------------------------------------------
**********Constant_Pool[6]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 26
--------------------------------------------------------------------
**********Constant_Pool[7]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 34
--------------------------------------------------------------------
**********Constant_Pool[8]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 40
--------------------------------------------------------------------
**********Constant_Pool[9]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 48
--------------------------------------------------------------------
**********Constant_Pool[10]=**********
9
##########Method##########
**********Classindex**********
0 5
**********Name&TypeIndex**********
0 17
--------------------------------------------------------------------
**********Constant_Pool[11]=**********
10
##########MethodRef##########
**********Classindex**********
0 6
**********Name&TypeIndex**********
0 18
--------------------------------------------------------------------
**********Constant_Pool[12]=**********
10
##########MethodRef##########
**********Classindex**********
0 6
**********Name&TypeIndex**********
0 19
--------------------------------------------------------------------
**********Constant_Pool[13]=**********
10
```

```
##########MethodRef##########
**********Classindex**********
0 6
**********Name&TypeIndex**********
0 20
--------------------------------------------------------------------
**********Constant_Pool[14]=**********
10
##########MethodRef##########
**********Classindex**********
0 7
**********Name&TypeIndex**********
0 21
--------------------------------------------------------------------
**********Constant_Pool[15]=**********
10
##########MethodRef##########
**********Classindex**********
0 8
**********Name&TypeIndex**********
0 18
--------------------------------------------------------------------
**********Constant_Pool[16]=**********
10
##########MethodRef##########
**********Classindex**********
0 9
**********Name&TypeIndex**********
0 22
--------------------------------------------------------------------
**********Constant_Pool[17]=**********
12
##########name&Type##########
**********Nameindex**********
0 24
**********SignatureIndex**********
0 25
--------------------------------------------------------------------
**********Constant_Pool[18]=**********
12
##########name&Type##########
**********Nameindex**********
0 27
**********SignatureIndex**********
0 28
--------------------------------------------------------------------
**********Constant_Pool[19]=**********
12
##########name&Type##########
**********Nameindex**********
0 30
**********SignatureIndex**********
0 31
--------------------------------------------------------------------
**********Constant_Pool[20]=**********
12
##########name&Type##########
**********Nameindex**********
0 32
```

```
**********SignatureIndex**********
0 33
--------------------------------------------------------------------
**********Constant_Pool[21]=**********
12
##########name&Type##########
**********Nameindex**********
0 35
**********SignatureIndex**********
0 36
--------------------------------------------------------------------
**********Constant_Pool[22]=**********
12
##########name&Type##########
**********Nameindex**********
0 49
**********SignatureIndex**********
0 50
--------------------------------------------------------------------
**********Constant_Pool[23]=**********
1
##########Constant Utf8##########
**********length**********
0 16
Length is 16
##########Actual String##########
java/lang/System
--------------------------------------------------------------------
**********Constant_Pool[24]=**********
1
##########Constant Utf8##########
**********length**********
0 3
Length is 3
##########Actual String##########
out
--------------------------------------------------------------------
**********Constant_Pool[25]=**********
1
##########Constant Utf8##########
**********length**********
0 21
Length is 21
##########Actual String##########
Ljava/io/PrintStream;
--------------------------------------------------------------------
**********Constant_Pool[26]=**********
1
##########Constant Utf8##########
**********length**********
0 22
Length is 22
##########Actual String##########
java/lang/StringBuffer
--------------------------------------------------------------------
**********Constant_Pool[27]=**********
1
##########Constant Utf8##########
**********length**********
```

```
0 6
Length is 6
##########Actual String##########
<init>
---------------------------------------------------------------------
**********Constant_Pool[28]=**********
1
##########Constant Utf8##########
**********length**********
0 3
Length is 3
##########Actual String##########
()V
---------------------------------------------------------------------
**********Constant_Pool[29]=**********
1
##########Constant Utf8##########
**********length**********
0 5
Length is 5
##########Actual String##########
hello
---------------------------------------------------------------------
**********Constant_Pool[30]=**********
1
##########Constant Utf8##########
**********length**********
0 6
Length is 6
##########Actual String##########
append
---------------------------------------------------------------------
**********Constant_Pool[31]=**********
1
##########Constant Utf8##########
**********length**********
0 44
Length is 44
##########Actual String##########
(Ljava/lang/String;)Ljava/lang/StringBuffer;
---------------------------------------------------------------------
**********Constant_Pool[32]=**********
1
##########Constant Utf8##########
**********length**********
0 8
Length is 8
##########Actual String##########
toString
---------------------------------------------------------------------
**********Constant_Pool[33]=**********
1
##########Constant Utf8##########
**********length**********
0 20
Length is 20
##########Actual String##########
()Ljava/lang/String;
---------------------------------------------------------------------
```

```
**********Constant_Pool[34]=**********
1
##########Constant Utf8##########
**********length**********
0 19
Length is 19
##########Actual String##########
java/io/PrintStream
--------------------------------------------------------------------
**********Constant_Pool[35]=**********
1
##########Constant Utf8##########
**********length**********
0 7
Length is 7
##########Actual String##########
println
--------------------------------------------------------------------
**********Constant_Pool[36]=**********
1
##########Constant Utf8##########
**********length**********
0 21
Length is 21
##########Actual String##########
(Ljava/lang/String;)V
--------------------------------------------------------------------
**********Constant_Pool[37]=**********
1
##########Constant Utf8##########
**********length**********
0 6
Length is 6
##########Actual String##########
hello1
--------------------------------------------------------------------
**********Constant_Pool[38]=**********
1
##########Constant Utf8##########
**********length**********
0 6
Length is 6
##########Actual String##########
hello2
--------------------------------------------------------------------
**********Constant_Pool[39]=**********
1
##########Constant Utf8##########
**********length**********
0 6
Length is 6
##########Actual String##########
hello3
--------------------------------------------------------------------
**********Constant_Pool[40]=**********
1
##########Constant Utf8##########
**********length**********
0 16
```

```
Length is 16
##########Actual String##########
java/lang/Object
--------------------------------------------------------------------
**********Constant_Pool[41]=**********
1
##########Constant Utf8##########
**********length**********
0 10
Length is 10
##########Actual String##########
test4.java
--------------------------------------------------------------------
**********Constant_Pool[42]=**********
1
##########Constant Utf8##########
**********length**********
0 13
Length is 13
##########Actual String##########
ConstantValue
--------------------------------------------------------------------
**********Constant_Pool[43]=**********
1
##########Constant Utf8##########
**********length**********
0 10
Length is 10
##########Actual String##########
Exceptions
--------------------------------------------------------------------
**********Constant_Pool[44]=**********
1
##########Constant Utf8##########
**********length**********
0 15
Length is 15
##########Actual String##########
LineNumberTable
--------------------------------------------------------------------
**********Constant_Pool[45]=**********
1
##########Constant Utf8##########
**********length**********
0 10
Length is 10
##########Actual String##########
SourceFile
--------------------------------------------------------------------
**********Constant_Pool[46]=**********
1
##########Constant Utf8##########
**********length**********
0 14
Length is 14
##########Actual String##########
LocalVariables
--------------------------------------------------------------------
**********Constant_Pool[47]=**********
```

```
1
##########Constant Utf8##########
**********length**********
0 4
Length is 4
##########Actual String##########
Code
----------------------------------------------------------------------
**********Constant_Pool[48]=**********
1
##########Constant Utf8##########
**********length**********
0 5
Length is 5
##########Actual String##########
test4
----------------------------------------------------------------------
**********Constant_Pool[49]=**********
1
##########Constant Utf8##########
**********length**********
0 4
Length is 4
##########Actual String##########
main
----------------------------------------------------------------------
**********Constant_Pool[50]=**********
1
##########Constant Utf8##########
**********length**********
0 22
Length is 22
##########Actual String##########
([Ljava/lang/String;)V
----------------------------------------------------------------------
**********access_flags**********
0 1
----------------------------------------------------------------------
**********this_class**********
0 9
----------------------------------------------------------------------
**********super_class**********
0 8
----------------------------------------------------------------------
**********interfaces_count**********
0 0
numberof interfaces 0
----------------------------------------------------------------------
----------------------------------------------------------------------
**********fields_count**********
0 0
numberof fieldscount 0
----------------------------------------------------------------------
----------------------------------------------------------------------
**********methods_count**********
0 2
numberof fieldscount 2
----------------------------------------------------------------------
&&&&&&&&&&Reading Methosds&&&&&&&&&&
```

```
---------------------------------------------------------------------
 Method no: 0
**********Access Flags**********
0 9
**********Name Index **********
0 49
**********Signature Index**********
0 50
**********AttributeCount**********
0 1
Attribute Count is1
---------------------------------------------------------------------
&&&&&&&&&&Reading Attributes&&&&&&&&&&
---------------------------------------------------------------------
Attribute no: 0
**********Attribute Name**********
0 47
**********Attribute Length **********
185
---------------------------------------------------------------------
&&&&&&&&&&Attribbutes&&&&&&&&&&
200
200
0
177
16
56
54
1
16
89
54
2
16
90
54
3
16
9
116
54
4
21
1
21
4
104
21
2
21
1
21
4
96
104
96
54
1
21
```

1
21
4
104
21
2
21
3
104
100
21
1
96
54
2
21
2
21
1
96
54
3
21
4
16
89
163
0
6
167
0
27
178
0
10
187
0
6
89
183
0
11
18
1
182
0
12
182
0
13
182
0
14
167
0
68
21
1
16

90
163
0
16
167
0
3
21
3
16
89
163
0
6
167
0
27
178
0
10
187
0
6
89
183
0
11
18
2
182
0
12
182
0
13
182
0
14
167
0
24
178
0
10
187
0
6
89
183
0
11
18
3
182
0
12
182
0
13

```
182
0
14
178
0
10
187
0
6
89
183
0
11
18
4
182
0
12
182
0
13
182
0
14
177
0
0
0
0
--------------------------------------------------------------------
--------------------------------------------------------------------
 Method no: 1
**********Access Flags**********
0 1
**********Name Index **********
0 27
**********Signature Index**********
0 28
**********AttributeCount**********
0 1
Attribute Count is1
--------------------------------------------------------------------
&&&&&&&&&&Reading Attributes&&&&&&&&&&
--------------------------------------------------------------------
Attribute no: 0
**********Attribute Name**********
0 47
**********Attribute Length **********
14
--------------------------------------------------------------------
&&&&&&&&&&Attribbutes&&&&&&&&&&
200
200
0
6
25
0
183
0
```

```
15
177
0
0
0
0
---------------------------------------------------------------------
```

4.5 Example 5: test5.java

Source Code:

```
public class test5 {
public static int fact (int i)
  { if (i ==1) return(1);
     else return(i*fact(i-1));
 }

  public static void main(String argv[]) {
    int a = 56; int c= 89; int d =90;
    int b;
     b = 7;
    System.out.println("factorial of  " + b + " is = "+ fact(b));

}

}
```

Intermediate Assembler Code:

```
Compiled from test5.java
modifiers class test5 extends java.lang.Object {
             int fact

             modifiers returnType fact (I);
             modifiers returnType main ();
             <init>()V;

Method fact
             0 iload_0
             2 bipush 1
             4 if_icmpeq 10
             7 goto 16
             10 bipush 1
             12 ireturn
             13 goto 28
             16 iload_0
             18 iload_0
             20 bipush 1
             22 isub
             23 invokestatic #0 <Method test5.fact(I)>
             26 imul
             27 ireturn
Method main
             0 bipush 7
             2 istore_4
             4 getstatic # <Field java.lang.System.out Ljava/io/PrintStream;>
             7 new # <Class java.lang.StringBuffer>
             10 dup
```

```
            11 invokenonvirtual # <Method java.lang.StringBuffer.<init>()V>
            14 ldc # <String "factorial of  ">
            16 invokevirtual # <Method java.lang.StringBuffer.append(Ljava/lang/
String;)Ljava/lang/StringBuffer;>
            19 iload_4
            21 invokevirtual # <Method java.lang.StringBuffer.append(I)Ljava/lang/
StringBuffer;> ==>0
            24 ldc # <String " is = ">
            26 invokevirtual # <Method java.lang.StringBuffer.append(Ljava/lang/
String;)Ljava/lang/StringBuffer;>
            29 iload_4
            31 invokestatic #1 <Method test5.fact(I)I>
            34 invokevirtual # <Method java.lang.StringBuffer.append(I)Ljava/lang/
StringBuffer;> ==>0
            37 invokevirtual # <Method java.lang.StringBuffer.toString()Ljava/lang/
String;>
            40 invokevirtual # <Method java.io.PrintStream.println(Ljava/lang/
String;)V>
            43 return

Method <init>
            0 aload_0
            2 invokenonvirtual # <Method java.lang.Object.<init>()V>
            5 return
}
```

***Interpreted Bytecode*:**

```
---------------------------------------------------------------------
**********magic**********
ca fe ba be
---------------------------------------------------------------------
**********minor_version**********
0 2
---------------------------------------------------------------------
**********major_version**********
0 45
---------------------------------------------------------------------
**********Constant_Pool_Count**********
0 54
##########Constant_Pool_Start##########
---------------------------------------------------------------------
**********Constant_Pool[1]=**********
8
##########String##########
**********Stringindex**********
0 31
---------------------------------------------------------------------
**********Constant_Pool[2]=**********
8
##########String##########
**********Stringindex**********
0 35
---------------------------------------------------------------------
**********Constant_Pool[3]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
```

```
0 25
---------------------------------------------------------------------
**********Constant_Pool[4]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 28
---------------------------------------------------------------------
**********Constant_Pool[5]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 38
---------------------------------------------------------------------
**********Constant_Pool[6]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 41
---------------------------------------------------------------------
**********Constant_Pool[7]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 49
---------------------------------------------------------------------
**********Constant_Pool[8]=**********
9
##########Method##########
**********Classindex**********
0 3
**********Name&TypeIndex**********
0 17
---------------------------------------------------------------------
**********Constant_Pool[9]=**********
10
##########MethodRef##########
**********Classindex**********
0 4
**********Name&TypeIndex**********
0 18
---------------------------------------------------------------------
**********Constant_Pool[10]=**********
10
##########MethodRef##########
**********Classindex**********
0 4
**********Name&TypeIndex**********
0 19
---------------------------------------------------------------------
**********Constant_Pool[11]=**********
10
##########MethodRef##########
**********Classindex**********
0 4
```

```
**********Name&TypeIndex**********
0 20
-------------------------------------------------------------------
**********Constant_Pool[12]=**********
10
##########MethodRef##########
**********Classindex**********
0 4
**********Name&TypeIndex**********
0 21
-------------------------------------------------------------------
**********Constant_Pool[13]=**********
10
##########MethodRef##########
**********Classindex**********
0 5
**********Name&TypeIndex**********
0 22
-------------------------------------------------------------------
**********Constant_Pool[14]=**********
10
##########MethodRef##########
**********Classindex**********
0 6
**********Name&TypeIndex**********
0 18
-------------------------------------------------------------------
**********Constant_Pool[15]=**********
10
##########MethodRef##########
**********Classindex**********
0 7
**********Name&TypeIndex**********
0 23
-------------------------------------------------------------------
**********Constant_Pool[16]=**********
10
##########MethodRef##########
**********Classindex**********
0 7
**********Name&TypeIndex**********
0 24
-------------------------------------------------------------------
**********Constant_Pool[17]=**********
12
##########name&Type##########
**********Nameindex**********
0 26
**********SignatureIndex**********
0 27
-------------------------------------------------------------------
**********Constant_Pool[18]=**********
12
##########name&Type##########
**********Nameindex**********
0 29
**********SignatureIndex**********
0 30
-------------------------------------------------------------------
```

```
**********Constant_Pool[19]=**********
12
##########name&Type##########
**********Nameindex**********
0 32
**********SignatureIndex**********
0 33
--------------------------------------------------------------------
**********Constant_Pool[20]=**********
12
##########name&Type##########
**********Nameindex**********
0 32
**********SignatureIndex**********
0 34
--------------------------------------------------------------------
**********Constant_Pool[21]=**********
12
##########name&Type##########
**********Nameindex**********
0 36
**********SignatureIndex**********
0 37
--------------------------------------------------------------------
**********Constant_Pool[22]=**********
12
##########name&Type##########
**********Nameindex**********
0 39
**********SignatureIndex**********
0 40
--------------------------------------------------------------------
**********Constant_Pool[23]=**********
12
##########name&Type##########
**********Nameindex**********
0 50
**********SignatureIndex**********
0 51
--------------------------------------------------------------------
**********Constant_Pool[24]=**********
12
##########name&Type##########
**********Nameindex**********
0 52
**********SignatureIndex**********
0 53
--------------------------------------------------------------------
**********Constant_Pool[25]=**********
1
##########Constant Utf8##########
**********length**********
0 16
Length is 16
##########Actual String##########
java/lang/System
--------------------------------------------------------------------
**********Constant_Pool[26]=**********
1
```

```
##########Constant Utf8##########
**********length**********
0 3
Length is 3
##########Actual String##########
out
---------------------------------------------------------------------
**********Constant_Pool[27]=**********
1
##########Constant Utf8##########
**********length**********
0 21
Length is 21
##########Actual String##########
Ljava/io/PrintStream;
---------------------------------------------------------------------
**********Constant_Pool[28]=**********
1
##########Constant Utf8##########
**********length**********
0 22
Length is 22
##########Actual String##########
java/lang/StringBuffer
---------------------------------------------------------------------
**********Constant_Pool[29]=**********
1
##########Constant Utf8##########
**********length**********
0 6
Length is 6
##########Actual String##########
<init>
---------------------------------------------------------------------
**********Constant_Pool[30]=**********
1
##########Constant Utf8##########
**********length**********
0 3
Length is 3
##########Actual String##########
()V
---------------------------------------------------------------------
**********Constant_Pool[31]=**********
1
##########Constant Utf8##########
**********length**********
0 14
Length is 14
##########Actual String##########
factorial of
---------------------------------------------------------------------
**********Constant_Pool[32]=**********
1
##########Constant Utf8##########
**********length**********
0 6
Length is 6
##########Actual String##########
```

```
append
--------------------------------------------------------------------
**********Constant_Pool[33]=**********
1
##########Constant Utf8##########
**********length**********
0 44
Length is 44
##########Actual String##########
(Ljava/lang/String;)Ljava/lang/StringBuffer;
--------------------------------------------------------------------
**********Constant_Pool[34]=**********
1
##########Constant Utf8##########
**********length**********
0 27
Length is 27
##########Actual String##########
(I)Ljava/lang/StringBuffer;
--------------------------------------------------------------------
**********Constant_Pool[35]=**********
1
##########Constant Utf8##########
**********length**********
0 6
Length is 6
##########Actual String##########
 is =
--------------------------------------------------------------------
**********Constant_Pool[36]=**********
1
##########Constant Utf8##########
**********length**********
0 8
Length is 8
##########Actual String##########
toString
--------------------------------------------------------------------
**********Constant_Pool[37]=**********
1
##########Constant Utf8##########
**********length**********
0 20
Length is 20
##########Actual String##########
()Ljava/lang/String;
--------------------------------------------------------------------
**********Constant_Pool[38]=**********
1
##########Constant Utf8##########
**********length**********
0 19
Length is 19
##########Actual String##########
java/io/PrintStream
--------------------------------------------------------------------
**********Constant_Pool[39]=**********
1
##########Constant Utf8##########
```

```
**********length**********
0 7
Length is 7
##########Actual String##########
println
---------------------------------------------------------------------
**********Constant_Pool[40]=**********
1
##########Constant Utf8##########
**********length**********
0 21
Length is 21
##########Actual String##########
(Ljava/lang/String;)V
---------------------------------------------------------------------
**********Constant_Pool[41]=**********
1
##########Constant Utf8##########
**********length**********
0 16
Length is 16
##########Actual String##########
java/lang/Object
---------------------------------------------------------------------
**********Constant_Pool[42]=**********
1
##########Constant Utf8##########
**********length**********
0 10
Length is 10
##########Actual String##########
test5.java
---------------------------------------------------------------------
**********Constant_Pool[43]=**********
1
##########Constant Utf8##########
**********length**********
0 13
Length is 13
##########Actual String##########
ConstantValue
---------------------------------------------------------------------
**********Constant_Pool[44]=**********
1
##########Constant Utf8##########
**********length**********
0 10
Length is 10
##########Actual String##########
Exceptions
---------------------------------------------------------------------
**********Constant_Pool[45]=**********
1
##########Constant Utf8##########
**********length**********
0 15
Length is 15
##########Actual String##########
LineNumberTable
```

```
-------------------------------------------------------------------
**********Constant_Pool[46]=**********
1
##########Constant Utf8##########
**********length**********
0 10
Length is 10
##########Actual String##########
SourceFile
-------------------------------------------------------------------
**********Constant_Pool[47]=**********
1
##########Constant Utf8##########
**********length**********
0 14
Length is 14
##########Actual String##########
LocalVariables
-------------------------------------------------------------------
**********Constant_Pool[48]=**********
1
##########Constant Utf8##########
**********length**********
0 4
Length is 4
##########Actual String##########
Code
-------------------------------------------------------------------
**********Constant_Pool[49]=**********
1
##########Constant Utf8##########
**********length**********
0 5
Length is 5
##########Actual String##########
test5
-------------------------------------------------------------------
**********Constant_Pool[50]=**********
1
##########Constant Utf8##########
**********length**********
0 4
Length is 4
##########Actual String##########
fact
-------------------------------------------------------------------
**********Constant_Pool[51]=**********
1
##########Constant Utf8##########
**********length**********
0 4
Length is 4
##########Actual String##########
(I)I
-------------------------------------------------------------------
**********Constant_Pool[52]=**********
1
##########Constant Utf8##########
**********length**********
```

```
0 4
Length is 4
##########Actual String##########
main
---------------------------------------------------------------------
**********Constant_Pool[53]=**********
1
##########Constant Utf8##########
**********length**********
0 22
Length is 22
##########Actual String##########
([Ljava/lang/String;)V
---------------------------------------------------------------------
**********access_flags**********
0 1
---------------------------------------------------------------------
**********this_class**********
0 7
---------------------------------------------------------------------
**********super_class**********
0 6
---------------------------------------------------------------------
**********interfaces_count**********
0 0
numberof interfaces 0
---------------------------------------------------------------------
---------------------------------------------------------------------
**********fields_count**********
0 0
numberof fieldscount 0
---------------------------------------------------------------------
---------------------------------------------------------------------
**********methods_count**********
0 3
numberof fieldscount 3
---------------------------------------------------------------------
&&&&&&&&&&Reading Methosds&&&&&&&&&&
---------------------------------------------------------------------
 Method no: 0
**********Access Flags**********
0 9
**********Name Index **********
0 50
**********Signature Index**********
0 51
**********AttributeCount**********
0 1
Attribute Count is1
---------------------------------------------------------------------
&&&&&&&&&&Reading Attributes&&&&&&&&&&
---------------------------------------------------------------------
Attribute no: 0
**********Attribute Name**********
0 48
**********Attribute Length **********
36
---------------------------------------------------------------------
&&&&&&&&&&Attribbutes&&&&&&&&&&
```

```
200
200
0
28
21
0
16
1
159
0
6
167
0
9
16
1
172
167
0
15
21
0
21
0
16
1
100
184
0
15
104
172
0
0
0
0
--------------------------------------------------------------------
--------------------------------------------------------------------
 Method no: 1
**********Access Flags**********
0 9
**********Name Index **********
0 52
**********Signature Index**********
0 53
**********AttributeCount**********
0 1
Attribute Count is1
--------------------------------------------------------------------
&&&&&&&&&&Reading Attributes&&&&&&&&&&
--------------------------------------------------------------------
Attribute no: 0
**********Attribute Name**********
0 48
**********Attribute Length **********
52
--------------------------------------------------------------------
&&&&&&&&&&Attribbutes&&&&&&&&&&
200
```

```
200
0
44
16
7
54
4
178
0
8
187
0
4
89
183
0
9
18
1
182
0
10
21
4
182
0
11
18
2
182
0
10
21
4
184
0
15
182
0
11
182
0
12
182
0
13
177
0
0
0
0
--------------------------------------------------------------------
--------------------------------------------------------------------
 Method no: 2
**********Access Flags**********
0 1
**********Name Index **********
0 29
**********Signature Index**********
```

```
0 30
**********AttributeCount**********
0 1
Attribute Count is1
--------------------------------------------------------------------
&&&&&&&&&&Reading Attributes&&&&&&&&&&
--------------------------------------------------------------------
Attribute no: 0
**********Attribute Name**********
0 48
**********Attribute Length **********
14
--------------------------------------------------------------------
&&&&&&&&&&Attribbutes&&&&&&&&&&
200
200
0
6
25
0
183
0
14
177
0
0
0
0
--------------------------------------------------------------------
--------------------------------------------------------------------
**********attributes_count**********
0 0
numberof attributescount 0
--------------------------------------------------------------------
--------------------------------------------------------------------
```

4.6 Example 6: `HelloAgain.java  (applet)`

Source Code:

```
import java.awt.Graphics;
import java.awt.Font;
import java.awt.Color;

public class HelloAgain extends java.applet.Applet {
   public void init() {
      resize(600, 300);
   }
   public void paint(Graphics g) {
      Font f = new Font("TimesRoman", 1, 36);
      g.setFont(f);
      g.setColor(Color.red);
      g.drawString("Hello, world!", 5, 100);
   }
}
```

Intermediate Assembler Code:

```
Compiled from HelloAgain.java
modifiers class HelloAgain extends java/applet/Applet {

             modifiers returnType init ();
             modifiers returnType paint ();
             <init>()V;

Method init
             0 aload_0
             2 sipush 600
             5 sipush 300
             8 invokevirtual # <Method java.applet.Applet.resize(II)V>
             11 return
Method paint
             0 new # <Class java.awt.Font>
             3 dup
             4 ldc # <String "TimesRoman">
             6 bipush 1
             8 bipush 36
             10 invokenonvirtual # <Method java.awt.Font.<init>(Ljava/lang/
String;II)V>
             13 astore_0
             15 aload_1
             17 aload_0
             19 invokevirtual # <Method java.awt.Graphics.setFont(Ljava/awt/Font;)V>
             22 aload_1
             24 getstatic # <Field java.awt.Color.red Ljava/awt/Color;>
             27 invokevirtual # <Method java.awt.Graphics.setColor(Ljava/awt/Color;)V>
             30 aload_1
             32 ldc # <String "Hello, world!">
             34 bipush 5
             36 bipush 100
             38 invokevirtual # <Method java.awt.Graphics.drawString(Ljava/lang/
String;II)V>
             41 return

Method <init>
             0 aload_0
             2 invokenonvirtual # <Method java/applet/Applet.<init>()V>
             5 return
}
```

Interpreted Bytecode:

```
---------------------------------------------------------------------
**********magic**********
ca fe ba be
---------------------------------------------------------------------
**********minor_version**********
0 2
---------------------------------------------------------------------
**********major_version**********
0 45
---------------------------------------------------------------------
**********Constant_Pool_Count**********
0 55
##########Constant_Pool_Start##########
---------------------------------------------------------------------
```

```
**********Constant_Pool[1]=**********
8
##########String##########
**********Stringindex**********
0 32
----------------------------------------------------------------------
**********Constant_Pool[2]=**********
8
##########String##########
**********Stringindex**********
0 41
----------------------------------------------------------------------
**********Constant_Pool[3]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 26
----------------------------------------------------------------------
**********Constant_Pool[4]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 29
----------------------------------------------------------------------
**********Constant_Pool[5]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 33
----------------------------------------------------------------------
**********Constant_Pool[6]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 36
----------------------------------------------------------------------
**********Constant_Pool[7]=**********
7
##########ConstantClass##########
**********nameindex**********
 Look at Constant_Pool[nameindex]
0 51
----------------------------------------------------------------------
**********Constant_Pool[8]=**********
9
##########Method##########
**********Classindex**********
0 6
**********Name&TypeIndex**********
0 20
----------------------------------------------------------------------
**********Constant_Pool[9]=**********
10
##########MethodRef##########
**********Classindex**********
```

```
0 3
**********Name&TypeIndex**********
0 17
---------------------------------------------------------------------
**********Constant_Pool[10]=**********
10
##########MethodRef##########
**********Classindex**********
0 4
**********Name&TypeIndex**********
0 18
---------------------------------------------------------------------
**********Constant_Pool[11]=**********
10
##########MethodRef##########
**********Classindex**********
0 5
**********Name&TypeIndex**********
0 19
---------------------------------------------------------------------
**********Constant_Pool[12]=**********
10
##########MethodRef##########
**********Classindex**********
0 5
**********Name&TypeIndex**********
0 21
---------------------------------------------------------------------
**********Constant_Pool[13]=**********
10
##########MethodRef##########
**********Classindex**********
0 5
**********Name&TypeIndex**********
0 22
---------------------------------------------------------------------
**********Constant_Pool[14]=**********
10
##########MethodRef##########
**********Classindex**********
0 3
**********Name&TypeIndex**********
0 23
---------------------------------------------------------------------
**********Constant_Pool[15]=**********
10
##########MethodRef##########
**********Classindex**********
0 7
**********Name&TypeIndex**********
0 24
---------------------------------------------------------------------
**********Constant_Pool[16]=**********
10
##########MethodRef##########
**********Classindex**********
0 7
**********Name&TypeIndex**********
0 25
```

```
---------------------------------------------------------------------
**********Constant_Pool[17]=**********
12
##########name&Type##########
**********Nameindex**********
0 27
**********SignatureIndex**********
0 28
---------------------------------------------------------------------
**********Constant_Pool[18]=**********
12
##########name&Type##########
**********Nameindex**********
0 30
**********SignatureIndex**********
0 31
---------------------------------------------------------------------
**********Constant_Pool[19]=**********
12
##########name&Type##########
**********Nameindex**********
0 34
**********SignatureIndex**********
0 35
---------------------------------------------------------------------
**********Constant_Pool[20]=**********
12
##########name&Type##########
**********Nameindex**********
0 37
**********SignatureIndex**********
0 38
---------------------------------------------------------------------
**********Constant_Pool[21]=**********
12
##########name&Type##########
**********Nameindex**********
0 39
**********SignatureIndex**********
0 40
---------------------------------------------------------------------
**********Constant_Pool[22]=**********
12
##########name&Type##########
**********Nameindex**********
0 42
**********SignatureIndex**********
0 31
---------------------------------------------------------------------
**********Constant_Pool[23]=**********
12
##########name&Type##########
**********Nameindex**********
0 30
**********SignatureIndex**********
0 43
---------------------------------------------------------------------
**********Constant_Pool[24]=**********
12
```

```
##########name&Type##########
**********Nameindex**********
0 52
**********SignatureIndex**********
0 43
--------------------------------------------------------------------
**********Constant_Pool[25]=**********
12
##########name&Type##########
**********Nameindex**********
0 53
**********SignatureIndex**********
0 54
--------------------------------------------------------------------
**********Constant_Pool[26]=**********
1
##########Constant Utf8##########
**********length**********
0 18
Length is 18
##########Actual String##########
java/applet/Applet
--------------------------------------------------------------------
**********Constant_Pool[27]=**********
1
##########Constant Utf8##########
**********length**********
0 6
Length is 6
##########Actual String##########
resize
--------------------------------------------------------------------
**********Constant_Pool[28]=**********
1
##########Constant Utf8##########
**********length**********
0 5
Length is 5
##########Actual String##########
(II)V
--------------------------------------------------------------------
**********Constant_Pool[29]=**********
1
##########Constant Utf8##########
**********length**********
0 13
Length is 13
##########Actual String##########
java/awt/Font
--------------------------------------------------------------------
**********Constant_Pool[30]=**********
1
##########Constant Utf8##########
**********length**********
0 6
Length is 6
##########Actual String##########
<init>
--------------------------------------------------------------------
```

```
**********Constant_Pool[31]=**********
1
##########Constant Utf8##########
**********length**********
0 23
Length is 23
##########Actual String##########
(Ljava/lang/String;II)V
----------------------------------------------------------------------
**********Constant_Pool[32]=**********
1
##########Constant Utf8##########
**********length**********
0 10
Length is 10
##########Actual String##########
TimesRoman
----------------------------------------------------------------------
**********Constant_Pool[33]=**********
1
##########Constant Utf8##########
**********length**********
0 17
Length is 17
##########Actual String##########
java/awt/Graphics
----------------------------------------------------------------------
**********Constant_Pool[34]=**********
1
##########Constant Utf8##########
**********length**********
0 7
Length is 7
##########Actual String##########
setFont
----------------------------------------------------------------------
**********Constant_Pool[35]=**********
1
##########Constant Utf8##########
**********length**********
0 18
Length is 18
##########Actual String##########
(Ljava/awt/Font;)V
----------------------------------------------------------------------
**********Constant_Pool[36]=**********
1
##########Constant Utf8##########
**********length**********
0 14
Length is 14
##########Actual String##########
java/awt/Color
----------------------------------------------------------------------
**********Constant_Pool[37]=**********
1
##########Constant Utf8##########
**********length**********
0 3
```

```
Length is 3
##########Actual String##########
red
---------------------------------------------------------------------
**********Constant_Pool[38]=**********
1
##########Constant Utf8##########
**********length**********
0 16
Length is 16
##########Actual String##########
Ljava/awt/Color;
---------------------------------------------------------------------
**********Constant_Pool[39]=**********
1
##########Constant Utf8##########
**********length**********
0 8
Length is 8
##########Actual String##########
setColor
---------------------------------------------------------------------
**********Constant_Pool[40]=**********
1
##########Constant Utf8##########
**********length**********
0 19
Length is 19
##########Actual String##########
(Ljava/awt/Color;)V
---------------------------------------------------------------------
**********Constant_Pool[41]=**********
1
##########Constant Utf8##########
**********length**********
0 13
Length is 13
##########Actual String##########
Hello, world!
---------------------------------------------------------------------
**********Constant_Pool[42]=**********
1
##########Constant Utf8##########
**********length**********
0 10
Length is 10
##########Actual String##########
drawString
---------------------------------------------------------------------
**********Constant_Pool[43]=**********
1
##########Constant Utf8##########
**********length**********
0 3
Length is 3
##########Actual String##########
()V
---------------------------------------------------------------------
**********Constant_Pool[44]=**********
```

```
1
##########Constant Utf8##########
**********length**********
0 15
Length is 15
##########Actual String##########
HelloAgain.java
---------------------------------------------------------------------
**********Constant_Pool[45]=**********
1
##########Constant Utf8##########
**********length**********
0 13
Length is 13
##########Actual String##########
ConstantValue
---------------------------------------------------------------------
**********Constant_Pool[46]=**********
1
##########Constant Utf8##########
**********length**********
0 10
Length is 10
##########Actual String##########
Exceptions
---------------------------------------------------------------------
**********Constant_Pool[47]=**********
1
##########Constant Utf8##########
**********length**********
0 15
Length is 15
##########Actual String##########
LineNumberTable
---------------------------------------------------------------------
**********Constant_Pool[48]=**********
1
##########Constant Utf8##########
**********length**********
0 10
Length is 10
##########Actual String##########
SourceFile
---------------------------------------------------------------------
**********Constant_Pool[49]=**********
1
##########Constant Utf8##########
**********length**********
0 14
Length is 14
##########Actual String##########
LocalVariables
---------------------------------------------------------------------
**********Constant_Pool[50]=**********
1
##########Constant Utf8##########
**********length**********
0 4
Length is 4
```

```
##########Actual String##########
Code
---------------------------------------------------------------------
**********Constant_Pool[51]=**********
1
##########Constant Utf8##########
**********length**********
0 10
Length is 10
##########Actual String##########
HelloAgain
---------------------------------------------------------------------
**********Constant_Pool[52]=**********
1
##########Constant Utf8##########
**********length**********
0 4
Length is 4
##########Actual String##########
init
---------------------------------------------------------------------
**********Constant_Pool[53]=**********
1
##########Constant Utf8##########
**********length**********
0 5
Length is 5
##########Actual String##########
paint
---------------------------------------------------------------------
**********Constant_Pool[54]=**********
1
##########Constant Utf8##########
**********length**********
0 22
Length is 22
##########Actual String##########
(Ljava/awt/Graphics;)V
---------------------------------------------------------------------
**********access_flags**********
0 1
---------------------------------------------------------------------
**********this_class**********
0 7
---------------------------------------------------------------------
**********super_class**********
0 3
---------------------------------------------------------------------
**********interfaces_count**********
0 0
numberof interfaces 0
---------------------------------------------------------------------
---------------------------------------------------------------------
**********fields_count**********
0 0
numberof fieldscount 0
---------------------------------------------------------------------
---------------------------------------------------------------------
**********methods_count**********
```

```
0 3
numberof fieldscount 3
--------------------------------------------------------------------
&&&&&&&&&&Reading Methosds&&&&&&&&&&
--------------------------------------------------------------------
 Method no: 0
**********Access Flags**********
0 1
**********Name Index **********
0 52
**********Signature Index**********
0 43
**********AttributeCount**********
0 1
Attribute Count is1
--------------------------------------------------------------------
&&&&&&&&&&Reading Attributes&&&&&&&&&&
--------------------------------------------------------------------
Attribute no: 0
**********Attribute Name**********
0 50
**********Attribute Length **********
20
--------------------------------------------------------------------
&&&&&&&&&&Attribbutes&&&&&&&&&&
200
200
0
12
25
0
17
2
88
17
1
44
182
0
9
177
0
0
0
0
--------------------------------------------------------------------
--------------------------------------------------------------------
 Method no: 1
**********Access Flags**********
0 1
**********Name Index **********
0 53
**********Signature Index**********
0 54
**********AttributeCount**********
0 1
Attribute Count is1
--------------------------------------------------------------------
&&&&&&&&&&Reading Attributes&&&&&&&&&&
```

```
-------------------------------------------------------------------
Attribute no: 0
**********Attribute Name**********
0 50
**********Attribute Length **********
50
-------------------------------------------------------------------
&&&&&&&&&&Attribbutes&&&&&&&&&&
200
200
0
42
187
0
4
89
18
1
16
1
16
36
183
0
10
58
0
25
1
25
0
182
0
11
25
1
178
0
8
182
0
12
25
1
18
2
16
5
16
100
182
0
13
177
0
0
0
0
-------------------------------------------------------------------
```

```
-------------------------------------------------------------------
 Method no: 2
**********Access Flags**********
0 1
**********Name Index **********
0 30
**********Signature Index**********
0 43
**********AttributeCount**********
0 1
Attribute Count is1
-------------------------------------------------------------------
&&&&&&&&&&Reading Attributes&&&&&&&&&&
-------------------------------------------------------------------
Attribute no: 0
**********Attribute Name**********
0 50
**********Attribute Length **********
14
-------------------------------------------------------------------
&&&&&&&&&&Attribbutes&&&&&&&&&&
200
200
0
6
25
0
183
0
14
177
0
0
0
0
-------------------------------------------------------------------
-------------------------------------------------------------------
**********attributes_count**********
0 0
numberof attributescount 0
-------------------------------------------------------------------
-------------------------------------------------------------------
```

5. Conclusion

The Java compiler can sucessfully compiles an one-class file. It is able to hanlde:

1) Data Types: integer, float, one-dimentional integer array, one-dimentional float array, some of system-defined date types such as String, Graphics, Font.
2) Statements: local variable declarations, empty statement, if statement, for statement, return statement, assignment, post-increment, method call, allocation expression
3) eveluate expressions, some field-access (only for system-defined fields, e.g out, Color), reognized operators are '+', '-', '*', '/', '%', '<', '<=', '>', '>=', '==', '!='

The compiler also has the following limitations:

1) The compiler's symbol table does not handle scope, which means the methods in a

method may not have common variable names.

2) The optimizer can not handle field access operations.

3) The compiler does not implement type-checking.

start_unit
    CompilationUnit
Type
    PrimitiveType
    Name
    ArrayType
PrimitiveType
    blooean
    char
    byte
    short
    int
    long
    float
    double
ArrayType
    Type []
PackageName
    Identifier
    PackageName . Identifier
TypeSpecifier
    PrimitiveType
    Identifier
CompilationUnit
    PackageStatementopt ImportStatementSeqopt TypeDeclarationsopt

PackageStatement
    package PackageName ;
TypeDeclarations
    TypeDeclaration
    TypeDeclarations TypeDeclaration
TypeDeclaration
    ClassDeclaration
    InterfaceDeclaration
ImportStatement
    PackageImportStatement
    TypeImportOnDemandStatement
ImportStatementSeq
    ImportStatement
    ImportStatementSeq ImportStatement
PackageImportStatement
    import PackageName ;
TypeImportOnDemandStatement
    import PackageName . * ;
ClassDeclaration
    ClassModifiersopt class Identifier Superopt Interfacesopt ClassBody

ClassModifiers
    ClassModifier
    ClassModifiers ClassModifier
ClassModifier
    abstract
    final
    public
Super
    extends PrimitiveType
    extends QualifiedName
Interfaces
    implements TypeNameList
ClassBody
    { FieldDeclarations }
    { }
FieldDeclarations
    FieldDeclaration
    FieldDeclarations FieldDeclaration
FieldDeclaration
    VariableMethodConstructorModifieropt VariableMethodConstructorDeclaration
    StaticInitializer
VariableMethodConstructorDeclaration
    VariableMethodModifieropt VariableMethodDeclaration
    ConstructorDeclaration
VariableMethodDeclaration
    VariableOnlyModifier Type VariableDeclarator ;
    Type VariableOrMethodDeclaration
    MethodOnlyModifieropt void MethodDeclarator Throwsopt MethodBody
    MethodOnlyModifieropt Type MethodDeclarator Throwsopt MethodBody
VariableOrMethodDeclaration
    VariableDeclarator ;
    MethodDeclarator Throwsopt MethodBody
ConstructorDeclaration
    ConstructorDeclarator Throwsopt ConstructorBody
ConstructorDeclarator
    TID TLEFTPAREN ParameterListopt TRIGHTPAREN
VariableMethodConstructorModifier
    public
    private
    protected
VariableMethodModifier
    static
    final
    VariableMethodModifier static
    VariableMethodModifier final
VariableOnlyModifier

transient
volatile
MethodOnlyModifier
abstract
native
synchronized
VariableDeclarator
DeclaratorName
VariableDeclarator , DeclaratorName
DeclaratorName = VariableInitializer
VariableDeclarator , DeclaratorName = VariableInitializer
DeclaratorName
Identifier
DeclaratorName [ ]
VariableInitializer
Expression
{ ArrayInitializersopt ,opt }
ArrayInitializers
VariableInitializer
ArrayInitializers , VariableInitializer
MethodDeclarator
DeclaratorName (ParameterListopt)
MethodDeclarator [ ]
ParameterList
Parameter
ParameterList , Parameter
Parameter
Type DeclaratorName
Throws
throws TypeNameList
TypeNameList
PrimitiveType
QualifiedName
TypeNameList , PrimitiveType
TypeNameList , QualifiedName
MethodBody
Block
;
ConstructorBody
{ ExplicitConstructorCallStatement BlockBodyopt }
MethodBody
ExplicitConstructorCallStatement
ThisOrSuper ArgumentListOrEmpty
StaticInitializer
static Block
InterfaceDeclaration

publicopt interface Identifier ExtendsInterfacesopt InterfaceBody
abstract interface Identifier ExtendsInterfacesopt InterfaceBody

ExtendsInterfaces
extends PrimitiveType
extends QualifiedName
ExtendsInterfaces , PrimitiveType
ExtendsInterfaces , QualifiedName

InterfaceBody
{ FieldDeclarations }

Block
{ BlockBody }
{ }

BlockBody
LocalVariableDeclarationAndStatements

LocalVariableDeclarationAndStatements
LocalVariableDeclarationOrStatement
LocalVariableDeclarationAndStatements LocalVariableDeclarationOrStatement

LocalVariableDeclarationOrStatement
LocalVariableDeclarationStatement
Statement

LocalVariableDeclarationStatement
PrimitiveType VariableDeclarator
Identifier VariableDeclarator

Statement
EmptyStatement
Identifier : Statement
ExpressionStatement
SelectionStatement
IterationStatement
JumpStatement
GuardingStatement
Block

EmptyStatement
;

LabeledStatement
case Expression : BlockBodyopt
default : BlockBodyopt

ValueExpression
Assignment
IncrementOrDecrement PrimaryExpression
PostIncrementOrPostDecrement
MethodCall
AllocationExpression

CommaExpression
ValueExpression
CommaExpression , ValueExpression

ExpressionStatement
    CommaExpression ;
SelectionStatement
    if (Expression) Statement
    if (Expression) Statement else Statement
    switch (Expression) { LabeledStatementSequence }
LabeledStatementSequence
    LabeledStatement
    LabeledStatementSequence LabeledStatement
IterationStatement
    while (Expression) Statement
    do Statement while { Expression } ;
    for { ForInit Expressionopt ; CommaExpressionopt } Statement
ForInit
    ExpressionStatement
    LocalVariableDeclarationStatement ;
JumpStatement
    break Identifieropt ;
    continue Identifieropt ;
    return Expressionopt ;
    throw Expression ;
GuardingStatement
    synchronized (Expression) Statement
    try Block Finally
    try Block Catches
    try Block Catches Finally
Catches
    Catch
    Catches Catch
Catch
    catch (Argument) Block
Argument
    Identifier Identifier
Finally
    finally Block
PrimaryExpression
    Name
    NotJustName
NotJustName
    AllocationExpression
    ComplexPrimary
ComplexPrimary
    Literal
    ParenthesizedExpression
    ArrayAccess
    FieldAccess

MethodCall
ParenthesizedExpression
(Expression)
Name
QualifiedName
ThisOrSuper
null
ThisOrSuper
this
super
QualifiedName
Identifier
QualifiedName . Identifier
ArrayAccess
Name [Expression]
ComplexPrimary [Expression]
FieldAccess
PrimaryExpression . Identifier
MethodCall
Name ArgumentListOrEmpty
PrimaryExpression . Identifier ArgumentListOrEmpty
ArgumentListOrEmpty
(ArgumentListopt)
ArgumentList
Expression
ArgumentList , Expression
AllocationExpression
new PrimitiveType (ArgumentListopt)
new PrimitiveType DimExpressions Dimsopt
new QualifiedName (ArgumentListopt)
new QualifiedName DimExpressions Dimsopt
DimExpressions
DimExpression
DimExpressions DimExpression
DimExpression
[Expression]
Dims
[ ]
Dims [ ]
PostfixExpression
PostIncrementOrPostDecrement
PrimaryExpression
PostIncrementOrPostDecrement
PrimaryExpression IncrementOrDecrement
IncrementOrDecrement
++

```
        --
UnaryExpression
        PreIncrementOrPreDecrement
        PlusOrMinus UnaryExpression
        UnaryExpressionNotPlusMinus
PlusOrMinus
        +
        -
PreIncrementOrPreDecrement
        IncrementOrDecrement PrimaryExpression
UnaryExpressionNotPlusMinus
        PostfixExpression
        complement UnaryExpression
        ! UnaryExpression
        CastExpression
CastExpression
        ( PrimitiveType ) UnaryExpression
        ( Identifier ) UnaryExpressionNotPlusMinus
MultiplicativeExpression
        UnaryExpression
        MultiplicativeExpression * UnaryExpression
        MultiplicativeExpression / UnaryExpression
        MultiplicativeExpression % UnaryExpression
AdditiveExpression
        MultiplicativeExpression
        AdditiveExpression PlusOrMinus MultiplicativeExpression
ShiftExpression
        AdditiveExpression
        ShiftExpression <<  AdditiveExpression
        ShiftExpression >>  AdditiveExpression
        ShiftExpression >>> AdditiveExpression
RelationalExpression
        ShiftExpression
        RelationalExpression < ShiftExpression
        RelationalExpression > ShiftExpression
        RelationalExpression <= ShiftExpression
        RelationalExpression >= ShiftExpression
        RelationalExpression instanceof TypeSpecifier Dimsopt
EqualityExpression
        RelationalExpression
        EqualityExpression == RelationalExpression
        EqualityExpression != RelationalExpression
AndExpression
        EqualityExpression
        AndExpression & EqualityExpression
ExclusiveOrExpression
```

AndExpression
ExclusiveOrExpression ^ AndExpression
InclusiveOrExpression
ExclusiveOrExpression
InclusiveOrExpression | ExclusiveOrExpression
ConditionalAndExpression
InclusiveOrExpression
ConditionalAndExpression and InclusiveOrExpression
ConditionalOrExpression
ConditionalAndExpression
ConditionalOrExpression || ConditionalAndExpression
ConditionalExpression
ConditionalOrExpression
ConditionalOrExpression ? Expression : ConditionalExpression
AssignmentExpression
ConditionalExpression
Assignment
Assignment
UnaryExpression = AssignmentExpression
UnaryExpression *= AssignmentExpression
UnaryExpression /= AssignmentExpression
UnaryExpression %= AssignmentExpression
UnaryExpression += AssignmentExpression
UnaryExpression -= AssignmentExpression
UnaryExpression <<= AssignmentExpression
UnaryExpression >>= AssignmentExpression
UnaryExpression >>>= AssignmentExpression
UnaryExpression &= AssignmentExpression
UnaryExpression ^= AssignmentExpression
UnaryExpression |= AssignmentExpression
Expression
AssignmentExpression
Identifier
TID
Literal
TINTEGERLITERAL
TFLOATINGPOINTLITERAL
TBOOLEANLITERAL
TCHARACTERLITERAL
TSTRINGLITERAL

Note: For detailed information about the definition of TID, TINTEGERLITERAL, TFLOATINGPOINTLITERAL, TBOOLEANLITERAL, TCHARACTERLITERAL, and TSTRINGLITERAL, please refer to ***The Java Language Specification*** (*Version 1.0 Beta*).