2CS701 Compiler Construction

Lab-6 Task

Submitted by: Labdhi Sheth 18BCE101

Aim: To generate Three Address codes for the assignment statement.

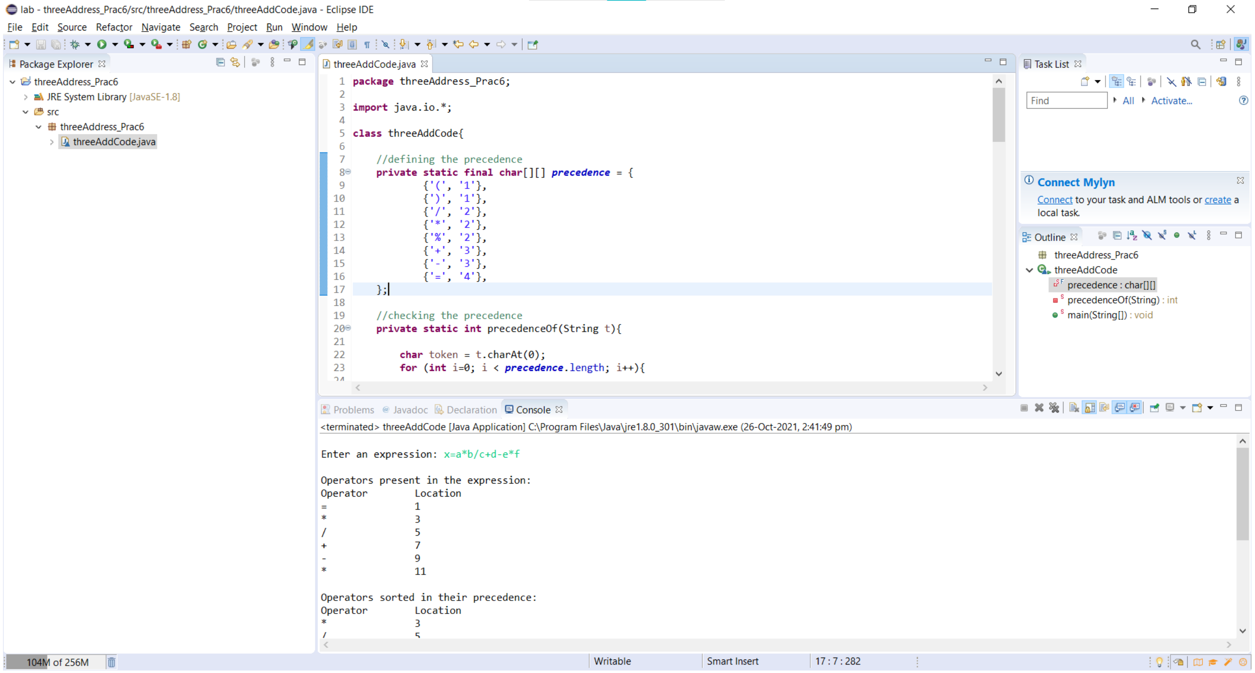
Java Code:

```
package threeAddress_Prac6;

import java.io.*;

class threeAddCode{

	//defining the precedence
	private static final char[][] precedence = {
		{'(', '1'},
		{')', '1'},
		{'/', '2'},
		{'*', '2'},
		{'%', '2'},
		{'+', '3'},
		{'-', '3'},
		{'=', '4'},

	};

	//checking the precedence
	private static int precedenceOf(String t){
        char token = t.charAt(0);
        for (int i=0; i < precedence.length; i++){

            if (token == precedence[i][0]){
                return Integer.parseInt(precedence[i][1]+"");
            }
        }
        return -1;
    }

    public static void main(String[] args) throws Exception     {

        int i, j, opc=0;
        char token;
        boolean processed[];

        String[][] operators = new String[10][2];
        String expr="", temp;
        BufferedReader in = new BufferedReader(new
InputStreamReader(System.in));
        System.out.print("\nEnter an expression: ");
        expr = in.readLine();
        processed = new boolean[expr.length()];

        for (i=0; i < processed.length; i++) {
            processed[i] = false;
        }

        for (i=0; i < expr.length(); i++){
            token = expr.charAt(i);
            for (j=0; j < precedence.length; j++){

                if (token==precedence[j][0]){
                    operators[opc][0] = token+"";
                    operators[opc][1] = i+"";
                    opc++;
                    break;
                }
            }
        }

        System.out.println("\nOperators present in the
expression:\nOperator\tLocation");
        for (i=0; i < opc; i++){
            System.out.println(operators[i][0] + "\t\t" +
operators[i][1]);
        }

        //sort
        for (i=opc-1; i >= 0; i--){
            for (j=0; j < i; j++){
                if (precedenceOf(operators[j][0]) >
precedenceOf(operators[j+1][0])) {
                    temp = operators[j][0];
                    operators[j][0] = operators[j+1][0];
                    operators[j+1][0] = temp;
                    temp = operators[j][1];
                    operators[j][1] = operators[j+1][1];
                    operators[j+1][1] = temp;
                }
            }
        }

        System.out.println("\nOperators sorted in their
precedence:\nOperator\tLocation");
        for (i=0; i < opc; i++){
            System.out.println(operators[i][0] + "\t\t" +
operators[i][1]);
        }
        System.out.println();

        System.out.println("\nThree Address code for the given
expression is as follows: ");

        for (i=0; i < opc; i++){
            j = Integer.parseInt(operators[i][1]+"");
            String op1="", op2="";
            if (processed[j-1]==true){
                if (precedenceOf(operators[i-1][0]) ==
precedenceOf(operators[i][0])){
                    op1 = "t"+i;
                }
                else{
                    for (int x=0; x < opc; x++){
                        if ((j-2) ==
Integer.parseInt(operators[x][1])){
                            op1 = "t"+(x+1)+"";
                        }
                    }
                }
            }
            else{
                op1 = expr.charAt(j-1)+"";
            }
            if (processed[j+1]==true){
                for (int x=0; x < opc; x++){
                    if ((j+2) ==
Integer.parseInt(operators[x][1])){
                        op2 = "t"+(x+1)+"";
                    }
                }
            }
            else{
                op2 = expr.charAt(j+1)+"";
            }

            System.out.println("t"+(i+1)+" =
"+op1+operators[i][0]+op2);
            processed[j] = processed[j-1] = processed[j+1] = true;
        }
    }
}
```

Output:

1. For input:

   x=a*b/c+d-e*f



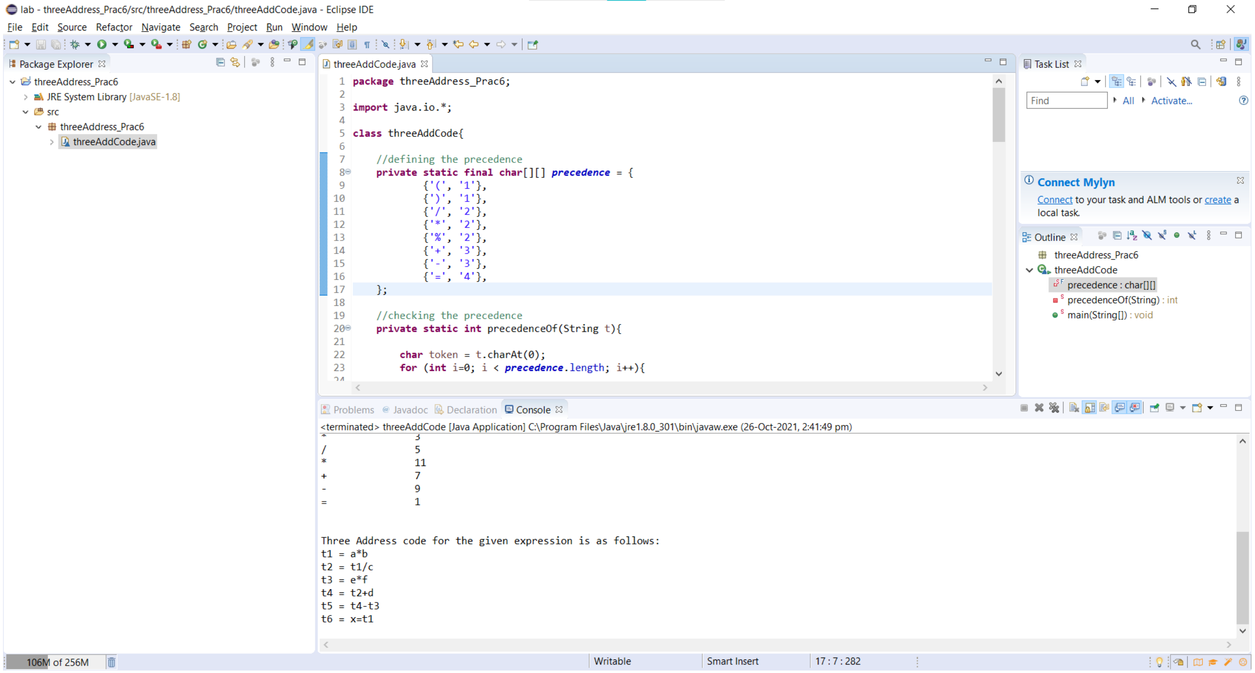
lab - threeAddress_Prac6/src/threeAddress_Prac6/threeAddCode.java - Eclipse IDE

threeAddCode.java

```
package threeAddress_Prac6;

import java.io.*;

class threeAddCode{

    //defining the precedence
    private static final char[][] precedence = {
            {'(', '1'},
            {')', '1'},
            {'/', '2'},
            {'*', '2'},
            {'%', '2'},
            {'+', '3'},
            {'-', '3'},
            {'=', '4'},
    };

    //checking the precedence
    private static int precedenceOf(String t){

        char token = t.charAt(0);
        for (int i=0; i < precedence.length; i++){
```

<terminated> threeAddCode [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (26-Oct-2021, 2:41:49 pm)

```
Enter an expression: x=a*b/c+d-e*f

Operators present in the expression:
Operator        Location
=               1
*               3
/               5
+               7
-               9
*               11

Operators sorted in their precedence:
Operator        Location
*               3
/               5
```



lab - threeAddress_Prac6/src/threeAddress_Prac6/threeAddCode.java - Eclipse IDE

threeAddCode.java

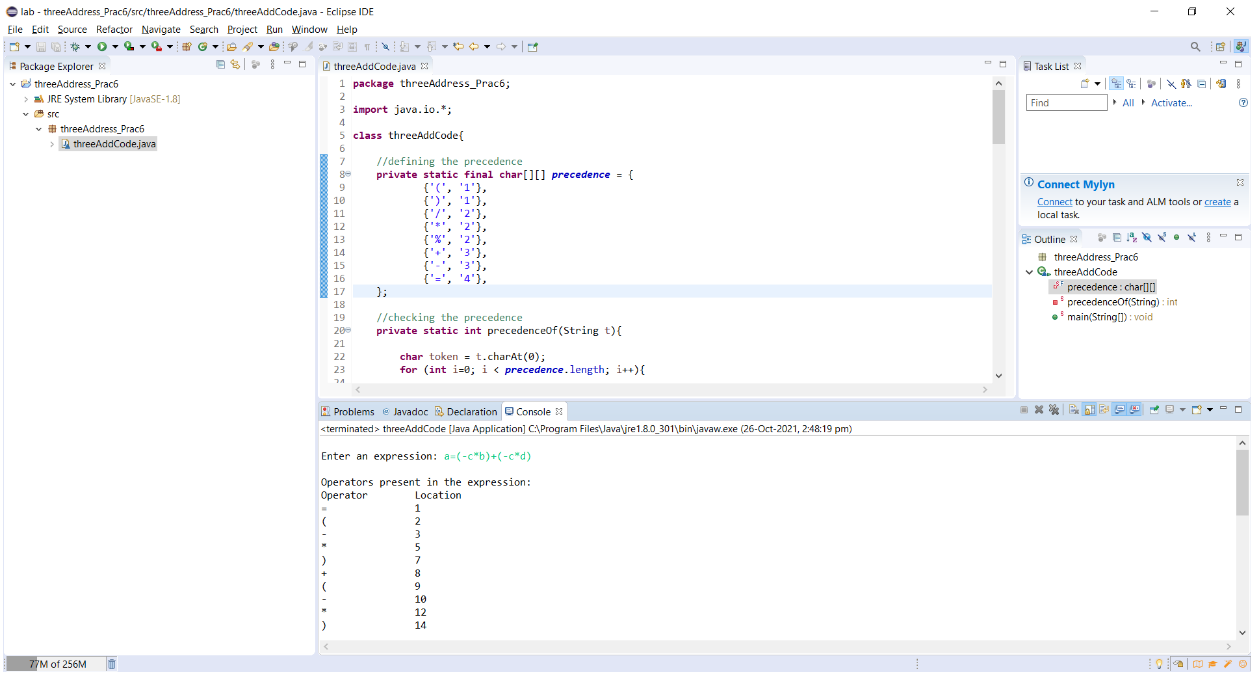
```
package threeAddress_Prac6;

import java.io.*;

class threeAddCode{

    //defining the precedence
    private static final char[][] precedence = {
            {'(', '1'},
            {')', '1'},
            {'/', '2'},
            {'*', '2'},
            {'%', '2'},
            {'+', '3'},
            {'-', '3'},
            {'=', '4'},
    };

    //checking the precedence
    private static int precedenceOf(String t){

        char token = t.charAt(0);
        for (int i=0; i < precedence.length; i++){
```

<terminated> threeAddCode [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (26-Oct-2021, 2:41:49 pm)

```
*               3
/               5
*               11
+               7
-               9
=               1


Three Address code for the given expression is as follows:
t1 = a*b
t2 = t1/c
t3 = e*f
t4 = t2+d
t5 = t4-t3
t6 = x=t1
```

2. For input:

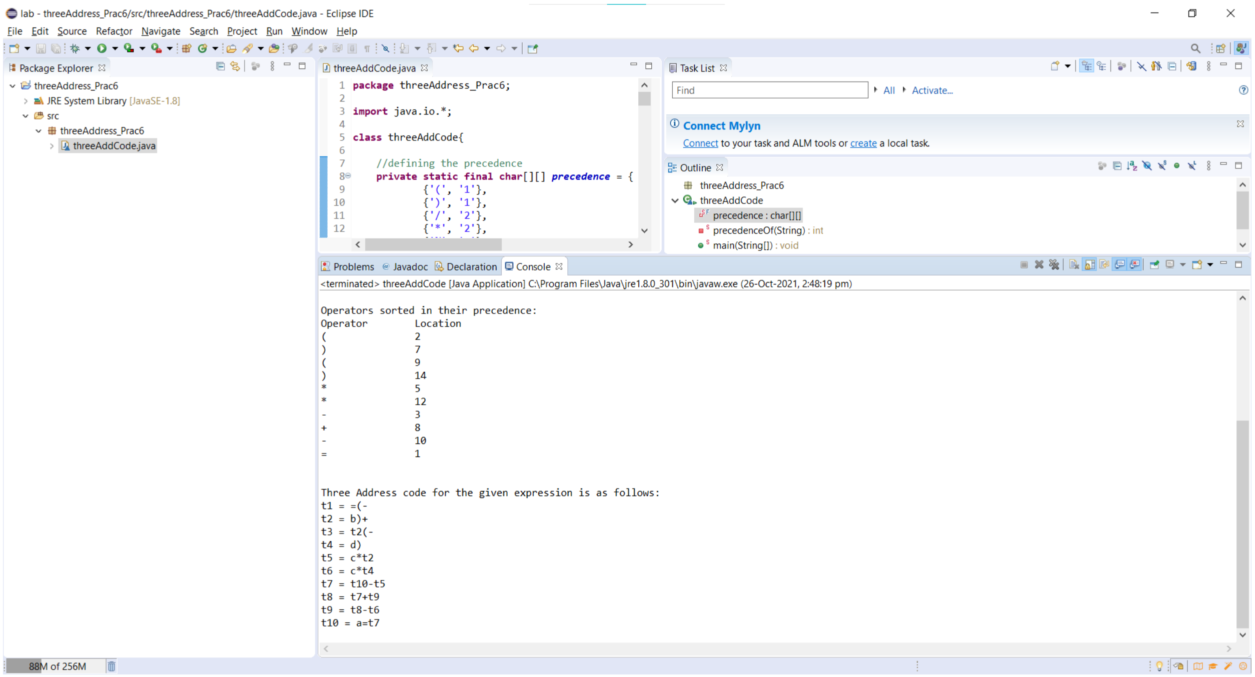
   a=(-c*b)+(-c*d)


lab - threeAddress_Prac6/src/threeAddress_Prac6/threeAddCode.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
threeAddress_Prac6
JRE System Library [JavaSE-1.8]
src
threeAddress_Prac6
threeAddCode.java
Task List
Find
All
Activate...
Connect Mylyn
Connect to your task and ALM tools or create a local task.
Outline
threeAddress_Prac6
threeAddCode
precedence : char[][]
precedenceOf(String) : int
main(String[]) : void
Problems
Javadoc
Declaration
Console
77M of 256M

threeAddCode.java

```java
package threeAddress_Prac6;

import java.io.*;

class threeAddCode{

    //defining the precedence
    private static final char[][] precedence = {
            {'(', '1'},
            {')', '1'},
            {'/', '2'},
            {'*', '2'},
            {'%', '2'},
            {'+', '3'},
            {'-', '3'},
            {'=', '4'},
    };

    //checking the precedence
    private static int precedenceOf(String t){

        char token = t.charAt(0);
        for (int i=0; i < precedence.length; i++){
```

```
<terminated> threeAddCode [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (26-Oct-2021, 2:48:19 pm)

Enter an expression: a=(-c*b)+(-c*d)

Operators present in the expression:
Operator        Location
=               1
(               2
-               3
*               5
)               7
+               8
(               9
-               10
*               12
)               14
```


lab - threeAddress_Prac6/src/threeAddress_Prac6/threeAddCode.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
threeAddress_Prac6
JRE System Library [JavaSE-1.8]
src
threeAddress_Prac6
threeAddCode.java
Task List
Find
All
Activate...
Connect Mylyn
Connect to your task and ALM tools or create a local task.
Outline
threeAddress_Prac6
threeAddCode
precedence : char[][]
precedenceOf(String) : int
main(String[]) : void
Problems
Javadoc
Declaration
Console
88M of 256M

threeAddCode.java

```java
package threeAddress_Prac6;

import java.io.*;

class threeAddCode{

    //defining the precedence
    private static final char[][] precedence = {
            {'(', '1'},
            {')', '1'},
            {'/', '2'},
            {'*', '2'},
```

```
<terminated> threeAddCode [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (26-Oct-2021, 2:48:19 pm)

Operators sorted in their precedence:
Operator        Location
(               2
)               7
(               9
)               14
*               5
*               12
-               3
+               8
-               10
=               1


Three Address code for the given expression is as follows:
t1 = =(-
t2 = b)+
t3 = t2(-
t4 = d)
t5 = c*t2
t6 = c*t4
t7 = t10-t5
t8 = t7+t9
t9 = t8-t6
t10 = a=t7
```

Conclusion:

The Three-address code is an intermediate code. It is used by the optimizing compilers. In the three-address code, the given expression is broken down into several separate instructions. These instructions can easily translate into assembly language. Each Three address code instruction has at most three operands. It is a combination of assignment and a binary operator.