**CODE OPTIMIZATION**

**INPUT:**

while ( i <= 100)

{

temp=54;

x = x + i ;

z = a + m ;

d = m ;

b = x + k ;

a = a + b ;

j = x + y ;

c = x + y + 20 ;

i++ ;

}

**CODE**

import java.io.\*;

import java.util.StringTokenizer;

class CodeOpt{

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

//read file

FileReader in = new FileReader("code.txt");

BufferedReader br = new BufferedReader(in);

String s,t;

StringTokenizer st;

String arr[] = new String[100];

String dependent[] = new String[100];

String d[][] = new String[100][10];

int count = 0; //total count

int index = 0; //count of dependent variables

while (( s=br.readLine() ) != null)

{

st= new StringTokenizer(s,"");

while(st.hasMoreTokens()) {

t = st.nextToken();

// System.out.println(t);

arr[count] = t;

// System.out.println(arr[count]);

d[count][0] = t;

d[count++][1]=""+0;

}

}

for(int i=0;i<count;i++){

String[] parts = arr[i].split(" ");

for(int k=0;k<parts.length;k++){

// System.out.println(parts[k]);

if(parts[k].compareToIgnoreCase("while")==0){

//System.out.println("ok");

k = k + 2;

dependent[index++] = parts[k];

}

else {

for(int z=0;z<index;z++){

if(dependent[z].compareToIgnoreCase(parts[k])==0)

{ dependent[index++]=arr[i].substring(0,1);

//System.out.println(dependent[index]);

}

}

}

}

}

/\* System.out.println("Dependent variables are");

for(int k=0;k<index;k++)

System.out.println(dependent[k]); \*/

for(int i=0;i<count;i++){

String[] part2 = d[i][0].split(" ");

for(int k=0;k<part2.length;k++){

for(int z=0;z<index;z++){

if((part2[k].substring(0,1)).compareToIgnoreCase("i")==0)

{

d[i][1]=""+ 1;

}

else

if(dependent[z].compareToIgnoreCase(part2[k])==0)

//dependent[index++]=arr[i].substring(0,1);

d[i][1]=""+ 1;

}

}

}

String rhs="";

for(int i=2;i<count;i++)

{

String temp=arr[i];

StringTokenizer st2=new StringTokenizer(arr[i],"=;");

String lhs=st2.nextToken();

if(st2.hasMoreTokens())

//System.out.println(st2.nextToken());

rhs=st2.nextToken();

for(int j=i+1;j<count;j++)

{

if(arr[j].contains(rhs))

{ //System.out.println(rhs);

d[j][0]=d[j][0].replace(rhs,lhs);

}

}

}

for(int i=0;i<count;i++){

if(d[i][1].compareToIgnoreCase("0")==1||d[i][0].compareToIgnoreCase("{")==0||d[i][0].compareToIgnoreCase("}")==0)

System.out.println(d[i][0]);

}

for(int i=0;i<count;i++){

if(d[i][1].compareToIgnoreCase("0")==0&&d[i][0].compareToIgnoreCase("{")!=0&&d[i][0].compareToIgnoreCase("}")!=0)

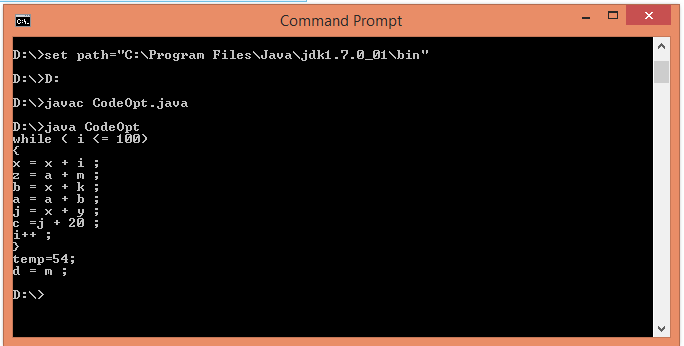
System.out.println(d[i][0]);

}

}

}

**OUTPUT:**

****