Sr. No.	Program	Remarks
1.	Introduction	
2.	Datatypes and Operators	
3.	Wrapper classes	
4.	Class and Object	
5.	Method Overloading	
6.	this and super keyword	
7.	Inheritance and Overriding	
8.	Exception Handling	
9.	Multithreading	
10.	Packages	
11.	Collection framework	
12.	Applets	
13.	Swing	
14.	JDBC	

PROGRAM 1

Introduction

Java is an object-oriented programming language developed by James Gosling and his team at Sun MicroSystems in 1995. It language was earlier known as Oak but later renamed as Java, from Java coffee of Indonesia.

Java was originally developed for cable TV remote controls but later it became popular in web technologies.

It was acquired by Oracle Corporation in 2010.

Editions of Java :

- Java Standard Edition(J2SE/JSE)
- Java Enterprise Edition(J2EE/JEE)
- Java Micro Edition(J2ME/JME)
- JavaFX

Versions of Java :

- The first public release of Java was JDK 1.0 in 1996.
- Some popular versions of Java are :

J2SE 5.0(2004)

Java SE 8(2014)

Java SE11(2018)

- Current stable version of Java is Java SE 17(2021)

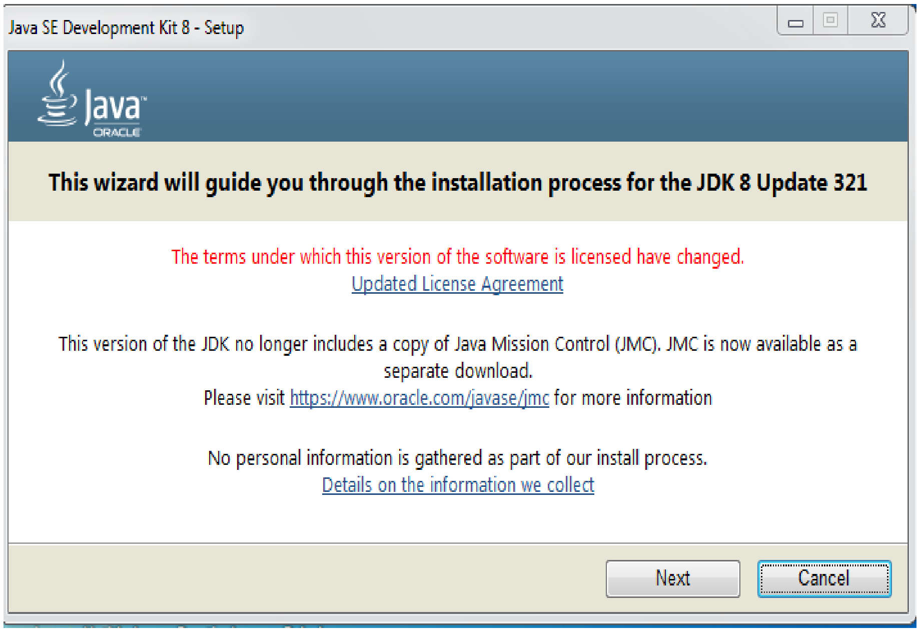
Features of Java :

- It is simple and secure language.
- It works on Write-Once-Run-Anywhere(WORA) principle.
- It is robust and safe.
- It supports various object-oriented concepts like encapsulation, inheritance, polymorphism, etc.
- It supports multithreaded programs.
- It is compiled and interpreted language.
- Java code is case-sensitive.
- It is a dynamic language. It also provides support for native methods.

Java Installation :

- ➢ Download the desired jdk version from : https://www.oracle.com/java/technologies/downloads/
- ➢ Follow the installation process:

Execute the downloaded file.



Java SE Development Kit 8 - Setup

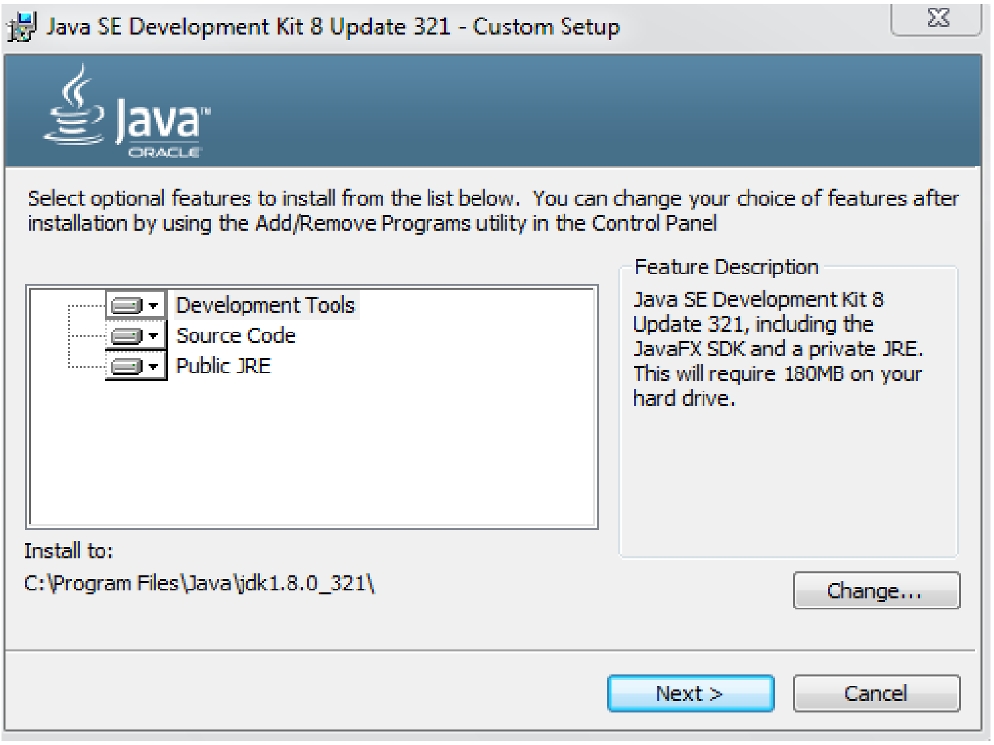
This wizard will guide you through the installation process for the JDK 8 Update 321

The terms under which this version of the software is licensed have changed.
Updated License Agreement

This version of the JDK no longer includes a copy of Java Mission Control (JMC). JMC is now available as a separate download.
Please visit https://www.oracle.com/javase/jmc for more information

No personal information is gathered as part of our install process.
Details on the information we collect

Click on next.



Java SE Development Kit 8 Update 321 - Custom Setup

Select optional features to install from the list below. You can change your choice of features after installation by using the Add/Remove Programs utility in the Control Panel

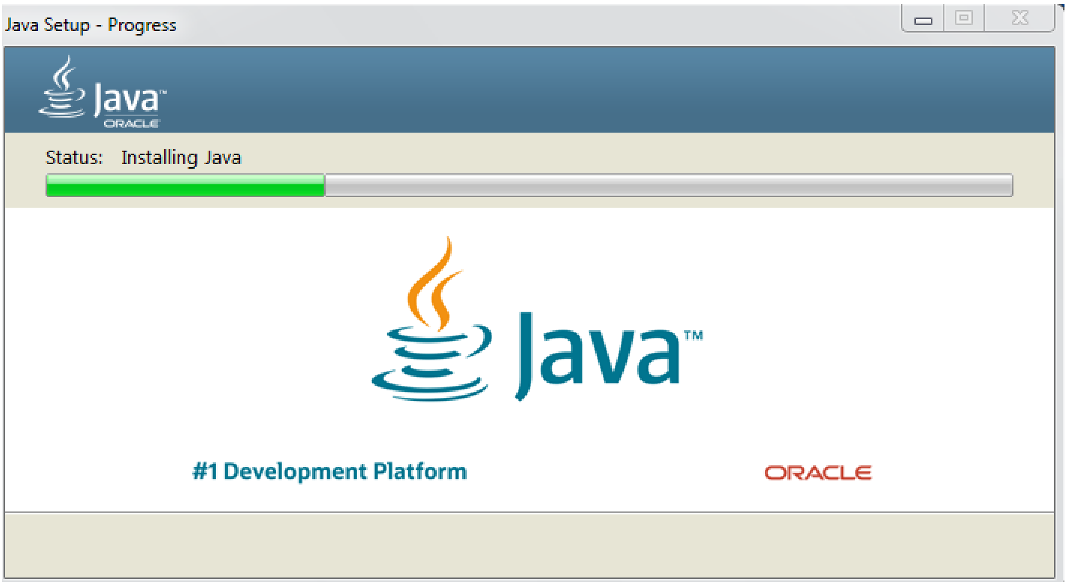
- Development Tools
- Source Code
- Public JRE

Feature Description

Java SE Development Kit 8 Update 321, including the JavaFX SDK and a private JRE. This will require 180MB on your hard drive.

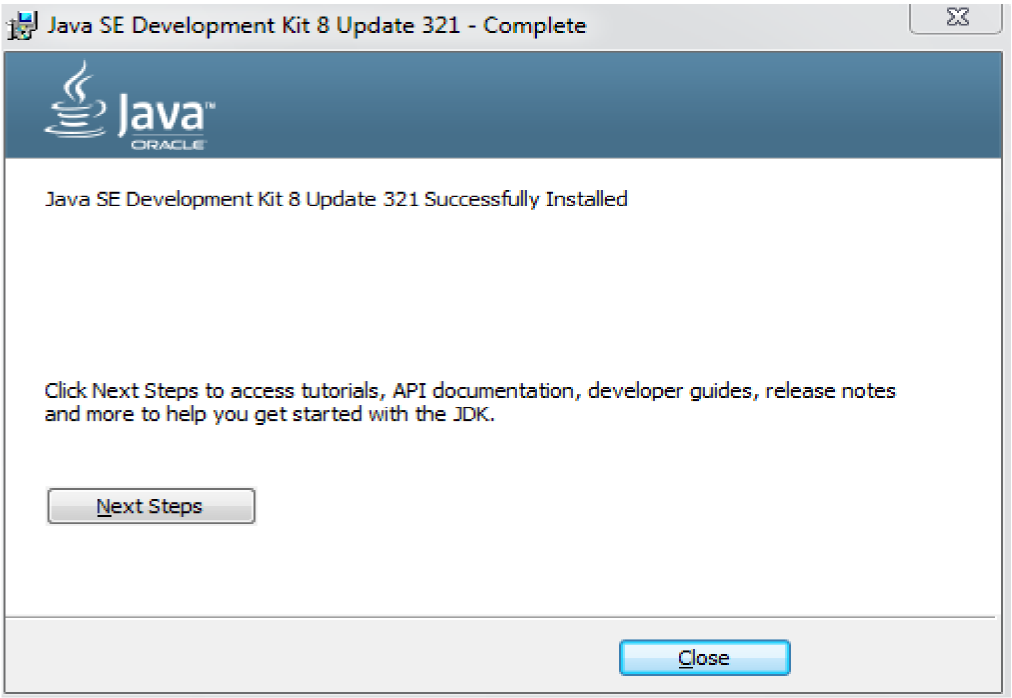
Install to:
C:\Program Files\Java\jdk1.8.0_321\

Choose the folder and click on next.



Java Setup - Progress

Status: Installing Java



Java SE Development Kit 8 Update 321 - Complete

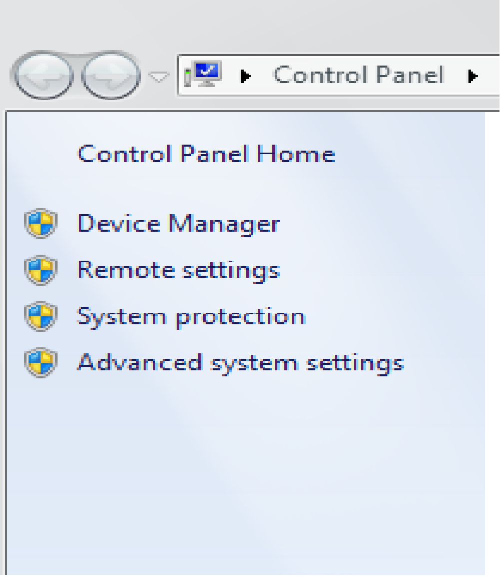
Java SE Development Kit 8 Update 321 Successfully Installed

Click Next Steps to access tutorials, API documentation, developer guides, release notes and more to help you get started with the JDK.

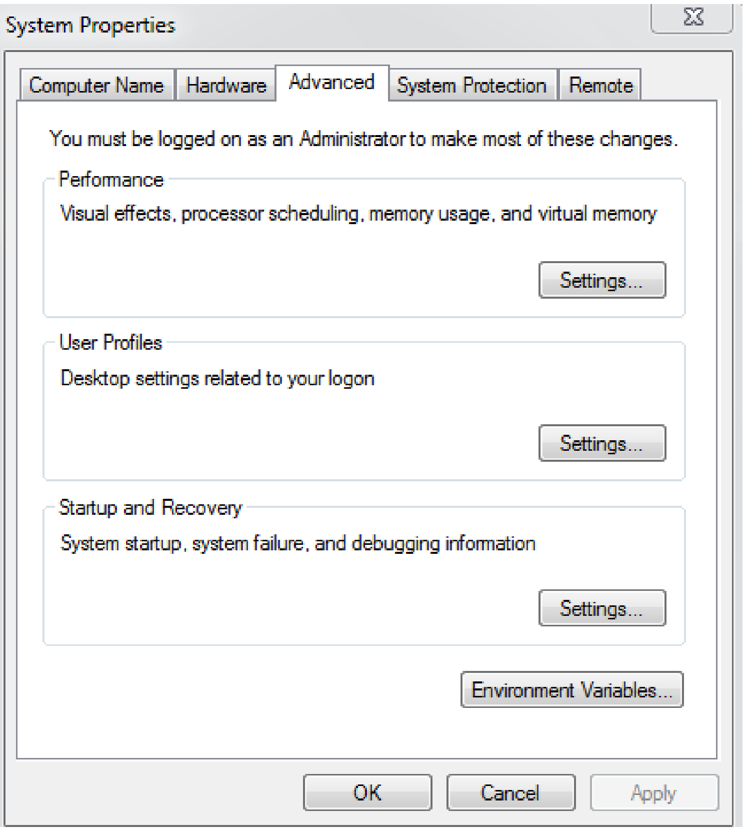
After installation close the window.

Setting the path variables :

- Right click on **My Computer-->properties.**



- Choose **Advanced system settings.**



You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

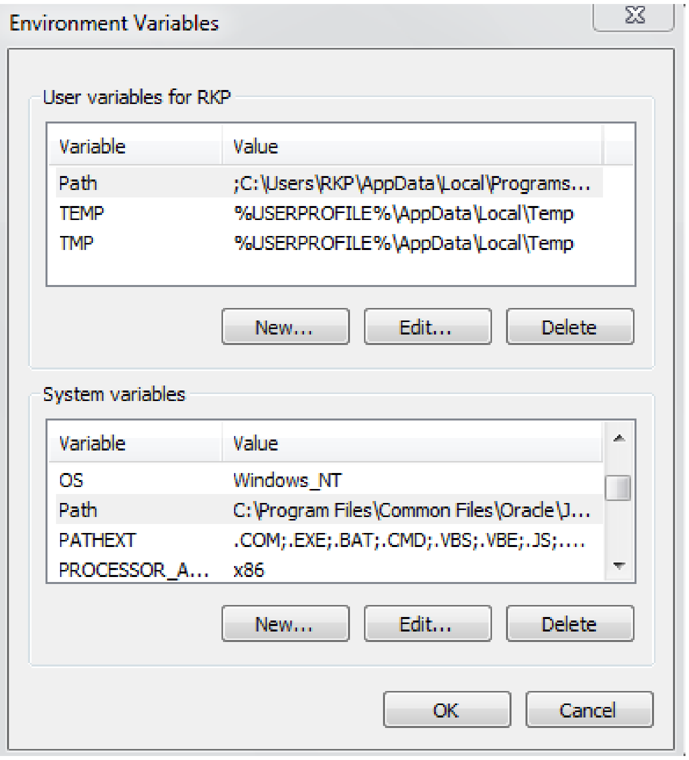
User Profiles

Desktop settings related to your logon

Startup and Recovery

System startup, system failure, and debugging information

- Click on **Environment Variables.**



Environment Variables

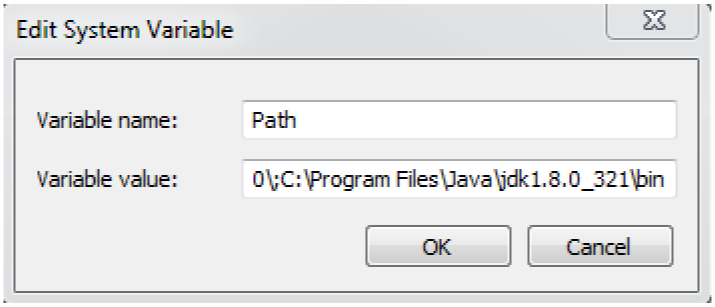
User variables for RKP

Variable	Value
Path	;C:\Users\RKP\AppData\Local\Programs...
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

System variables

Variable	Value
OS	Windows_NT
Path	C:\Program Files\Common Files\Oracle\J...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_A...	x86

- Copy the path of bin folder in path variable.



Edit System Variable

Variable name: Path

Variable value: 0\;C:\Program Files\Java\jdk1.8.0_321\bin

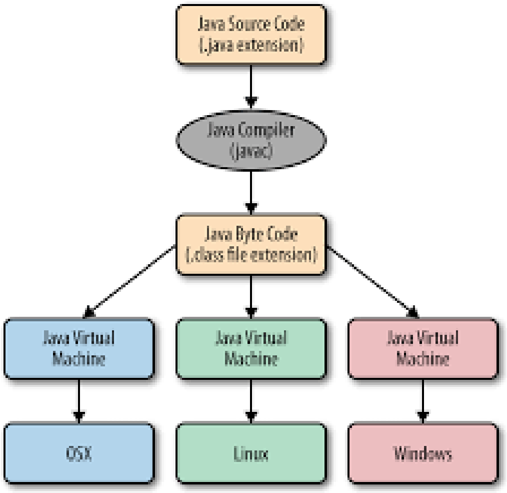
Open command prompt and run following command:

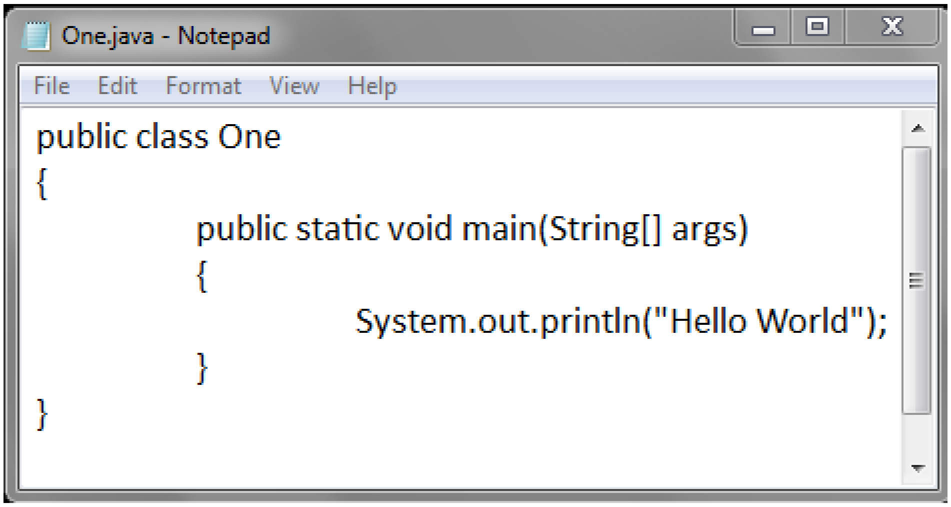
```
java –version
```

Java compilation and execution:

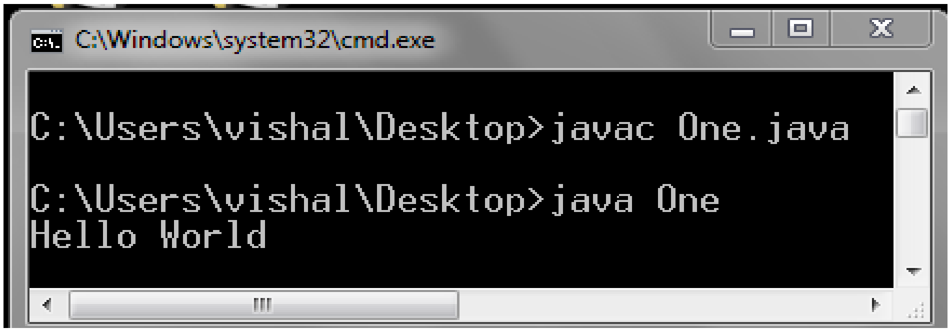
Java is a platform independent language. This is achieved using a special code-byte code and the platform specific interpreter-JVM.

Java programs are compiled into a machine independent code known as byte code(.class file). Byte code is then provided as an input to JVM which in return interprets and executes it. JVM stands for Java Virtual Machine which creates a virtual environment for execution byte code on different machines. JVM is specific to an operating system.





```
public class One
{
                public static void main(String[] args)
                {
                                System.out.println("Hello World");
                }
}
```



```
C:\Users\vishal\Desktop>javac One.java

C:\Users\vishal\Desktop>java One
Hello World
```

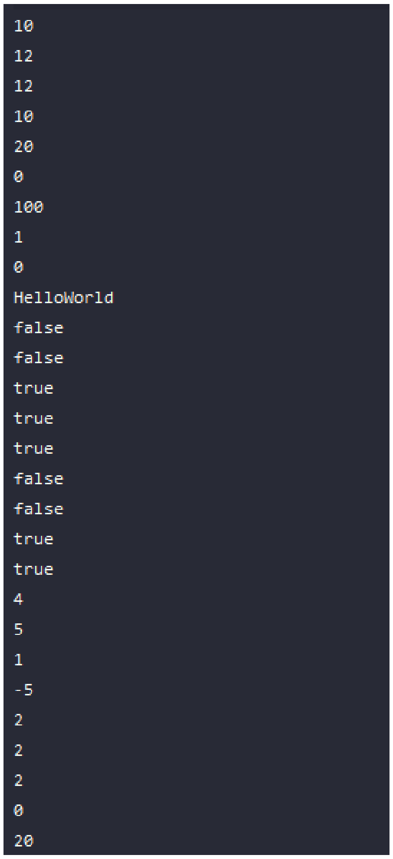
PROGRAM 2

Datatypes and Operators

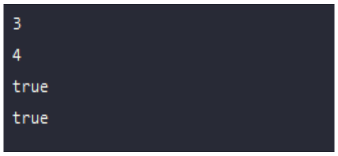
```
public class Basics {
  public static void main(String[] args) {

      /* DATATYPES */

      //PRIMITIVE DATATYPES

      //Numeric datatypes
      //Integer datatypes
      byte b=10;
      short s=10;
      int i=20;
      long l=30l;

      //floating point
      float f=10.2f;
      double d=20.54d;

      //non numeric datatypes
      char c=’a’;
      boolean bool=true;

      //REFERENCE DATATYPES


      String str=”hello”;
      StringBuffer sb=new StringBuffer(”world”);

      /* OPERATORS  */

      //increment/decrement
      int a=10;
      System.out.println(a++);    //post increment
      System.out.println(++a);    //pre increment
      System.out.println(a--);    //post decrement
      System.out.println(--a);    //pre decrement

      //arithmetic
      int x=6;
      int y=4;
```

```
System.out.println(a+b);    //addition
System.out.println(a-b);    //subtraction
System.out.println(a*b);    //multiplication
System.out.println(a/b);    //division
System.out.println(a%b);    //modulus

//String concatenation
System.out.println(”Hello”+”World”);

//Relational
int p=20;
int q=30;
System.out.println(a>b);    //greater than
System.out.println(a<b);    //less than
System.out.println(a<=b);    //less than equal to
System.out.println(a>=b);    //greater than equal to
System.out.println(a==b);    //equal to
System.out.println(a!=b);    //not equal to

//logical
System.out.println(a>b&&x==y);    //AND
System.out.println(a<b||x>y);    //OR
System.out.println(!(x==y));    //NOT

//bitwise
System.out.println(4&5);    //bitwise AND
System.out.println(4|5);    //bitwise OR
System.out.println(4^5);    //bitwise XOR
System.out.println(~4);    //bitwise complement

//shift
System.out.println(10>>2);    //right shift
System.out.println(10>>2);    //left shift
System.out.println(10>>>2);    //unsigned right shift

//assignment
int u=10;
int v=20;
u=v;

//shorthand assignment
u+=v;
u-=v;
u*=v;
u/=v;
u%=v;
System.out.println(u);
```

```
    System.out.println(v);

    //conditional/ ternary
    System.out.println(10>2?3:4);
    System.out.println(10<2?3:4);

    //instance of
    System.out.println(”abc” instanceof String);
    System.out.println(”abc” instanceof Object);
      }
}
```



```
10
12
12
10
20
0
100
1
0
HelloWorld
false
false
true
true
true
false
false
true
true
4
5
1
-5
2
2
2
0
20
```

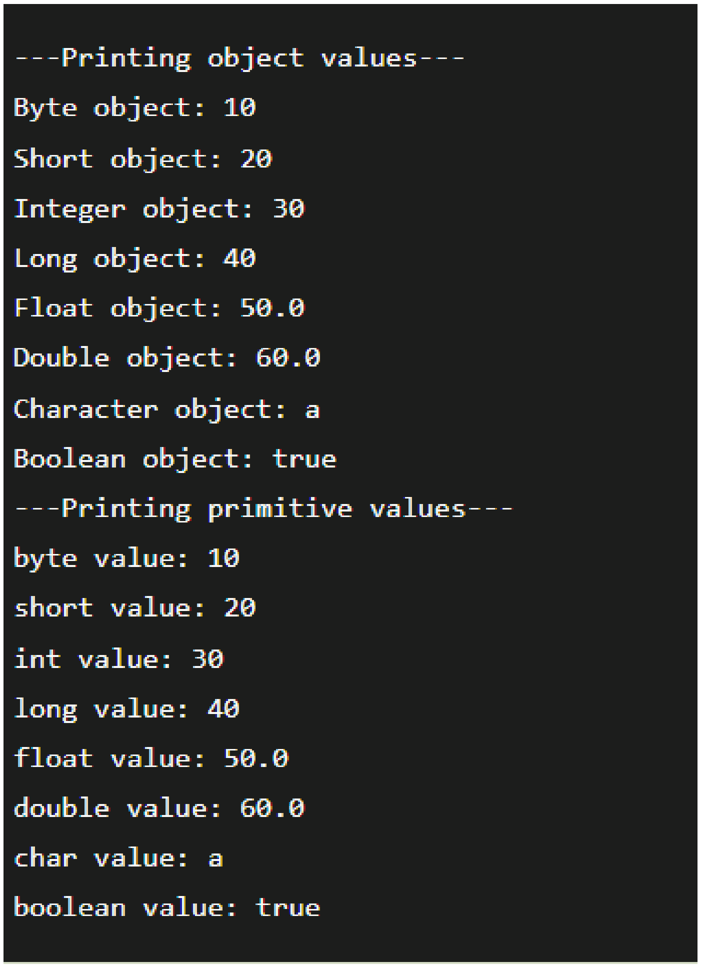


```
3
4
true
true
```

PROGRAM 3

Wrapper Classes

```
//Java Program to convert all primitives into its corresponding
//wrapper objects and vice-versa
public class WrapperExample3{
public static void main(String args[]){
byte b=10;
short s=20;
int i=30;
long l=40;
float f=50.0F;
double d=60.0D;
char c=’a’;
boolean b2=true;

//Autoboxing: Converting primitives into objects
Byte byteobj=b;
Short shortobj=s;
Integer intobj=i;
Long longobj=l;
Float floatobj=f;
Double doubleobj=d;
Character charobj=c;
Boolean boolobj=b2;

//Printing objects
System.out.println(”---Printing object values---”);
System.out.println(”Byte object: ”+byteobj);
System.out.println(”Short object: ”+shortobj);
System.out.println(”Integer object: ”+intobj);
System.out.println(”Long object: ”+longobj);
System.out.println(”Float object: ”+floatobj);
System.out.println(”Double object: ”+doubleobj);
System.out.println(”Character object: ”+charobj);
System.out.println(”Boolean object: ”+boolobj);

//Unboxing: Converting Objects to Primitives
byte bytevalue=byteobj;
short shortvalue=shortobj;
int intvalue=intobj;
long longvalue=longobj;
float floatvalue=floatobj;
double doublevalue=doubleobj;
char charvalue=charobj;
boolean boolvalue=boolobj;

//Printing primitives
System.out.println(”---Printing primitive values---”);
System.out.println(”byte value: ”+bytevalue);
System.out.println(”short value: ”+shortvalue);
System.out.println(”int value: ”+intvalue);
System.out.println(”long value: ”+longvalue);
System.out.println(”float value: ”+floatvalue);
System.out.println(”double value: ”+doublevalue);
System.out.println(”char value: ”+charvalue);
System.out.println(”boolean value: ”+boolvalue);
}}
```



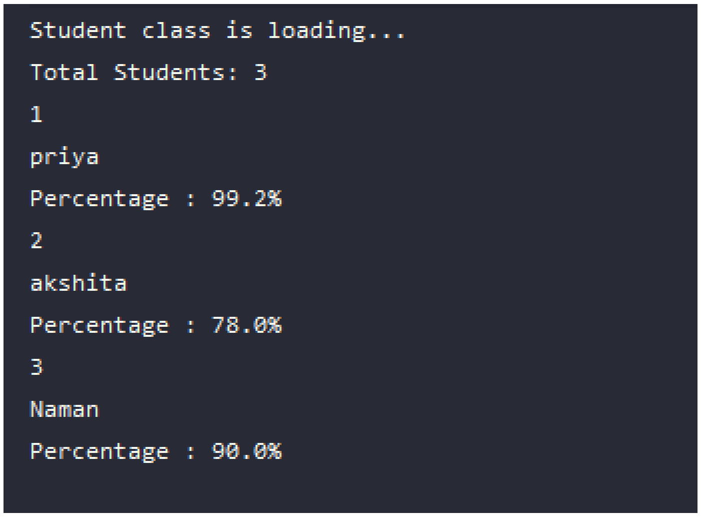
```
---Printing object values---
Byte object: 10
Short object: 20
Integer object: 30
Long object: 40
Float object: 50.0
Double object: 60.0
Character object: a
Boolean object: true
---Printing primitive values---
byte value: 10
short value: 20
int value: 30
long value: 40
float value: 50.0
double value: 60.0
char value: a
boolean value: true
```

PROGRAM 4

Class and Object

```
import java.util.ArrayList;

class Student
{
   //instance variables
   private int id;
   private String name;
   int marks;
   //static variable
   static final int maxMarks=500;
   static int count=0;
   //static block
   static
   {
      System.out.println(”Student class is loading...”);
   }
   //instance block
   {
      count++;
   }
   //contructor
   Student(int id,String name)
   {
      this.id=id;
      this.name=name;
   }
   //getters
   int getId()
   {
      return id;
   }
   String getName()
   {
      return name;
   }
   //setters
   void setId(int id)
   {
      this.id=id;
   }
```

```
    void setName(String name)
    {
        this.name=name;
    }

    void updateMarks(int marks)
    {
        this.marks=marks;
    }
    void displayMarks()
    {
        System.out.println(”Marks: ”+marks);
    }
    void displayPercentage()
    {
        double p=(double)marks/maxMarks*100;
        System.out.println(”Percentage : ”+p+”%”);
    }
}

class Test
{
    public static void main(String[] args) {
        ArrayList<Student> s=new ArrayList<>();
       s.add(new Student(1,”priya”));
        s.add(new Student(2,”akshita”));
        s.add(new Student(3,”Naman”));

        s.get(0).updateMarks(496);
        s.get(1).updateMarks(390);
        s.get(2).updateMarks(450);

        System.out.println(”Total Students: ”+Student.count);
        for(Student st:s)
        {
            System.out.println(st.getId());
            System.out.println(st.getName());
            st.displayPercentage();
        }

    }
}
```

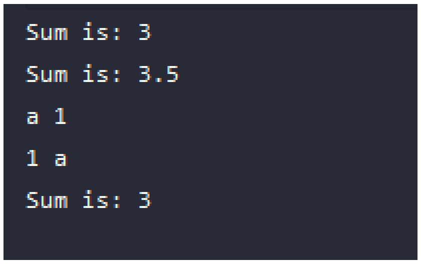


```
Student class is loading...
Total Students: 3
1
priya
Percentage : 99.2%
2
akshita
Percentage : 78.0%
3
Naman
Percentage : 90.0%
```

PROGRAM 5

Method Overloading

```
public class Sum {
    public static void sum(long a,long b)
    {
        long sum=a+b;
        System.out.println(”Sum is: ”+sum);
    }
    public static void sum(int a,int b,int c)
    {
        int sum=a+b+c;
        System.out.println(”Sum is: ”+sum);
    }
    public static void sum(float a,float b)
    {
        float sum=a+b;
        System.out.println(”Sum is: ”+sum);
    }
    public static void display(String s,int a)
    {
        System.out.println(s+” ”+a);
    }
    public static void display(int a,String s)
    {
        System.out.println(a+” ”+s);
    }

    public static void main(String[] args) {
        sum(1l,2l);
        sum(2.3f,1.2f);
        display(”a”,1);
        display(1,”a”);
        sum(1,2);          //type promotion

    }
}
```



```
Sum is: 3
Sum is: 3.5
a 1
1 a
Sum is: 3
```

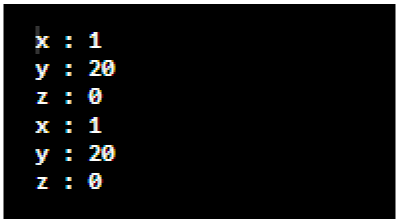
PROGRAM 6

Uses of this and super keyword

```
class B
{
   A a;
   B(A obj)
   {
      a=obj;
   }
   void display()
   {
      a.display();
   }
}


class P
{
   int x;
   P()
   {
      x=1;
   }
   P(int x)
   {
      this.x=x;
   }
   void display()
   {
      System.out.println("x : "+x);
   }
}

public class A extends P{
   int y,z;
   A()
```

```
{
   super();
}
A(int x)
{
   super(x);
}
A(int x,int y)
{
   //using this for invoking no-arg constructor
   this();
   //using this to refer to instance variables
   this.y=y;
}
A(int x,int y,int z)
{
   //using this to invoke parametrized constructor
   this(x,y);
   this.z=z;
}
public void displayY()
{
   System.out.println(”y : ”+y);
}
public void displayZ()
{
   System.out.println(”z : ”+z);
}
public void display()
{
   super.display();
   //System.out.println(super.x);
   //using this to invoke instance methods
   this.displayY();
   this.displayZ();
}
public A getA()
{
   return this;
}
```

```
 public B getB()
 {
    return new B(this);
 }
  public static void main(String args[]) {
   A a=new A(10,20);
   a.getA().display();

   a.getB().display();

 }
}
```

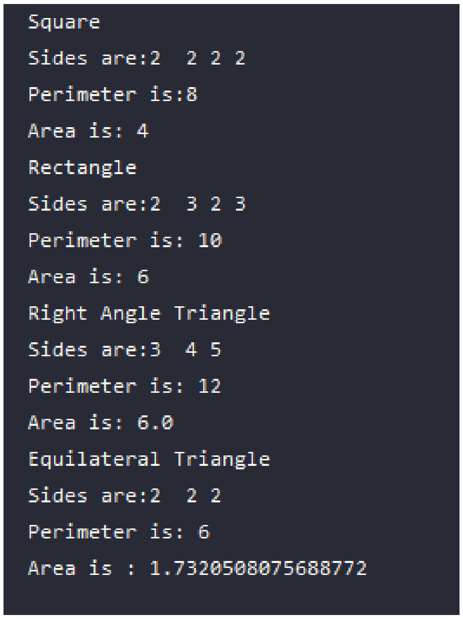


```
x : 1
y : 20
z : 0
x : 1
y : 20
z : 0
```

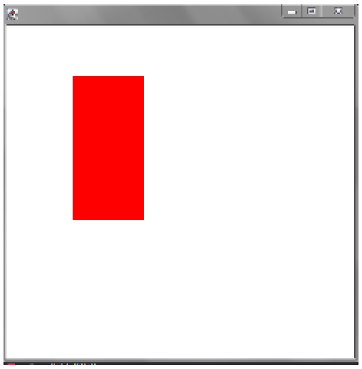
PROGRAM 7

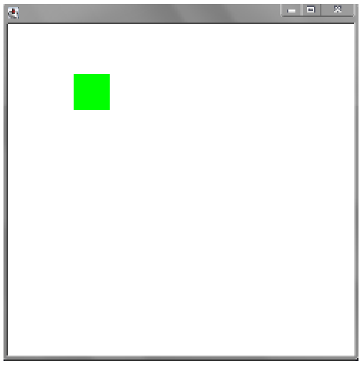
Inheritance and Overriding

```
import java.awt.*;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;


abstract public class Shape {
    int color;
    static final int r=1;
    static final int g=2;
    static final int b=3;
    Shape(int color)
    {
        this.color=color;
    }
    void displayColor()
    {
        System.out.println(”Color of the shape is: ”+color);
    }
    abstract void perimeter();
    abstract void area();
    abstract void displaySides();
    abstract void displayName();
}

abstract public class Quadrilateral extends Shape{
    static final int noOfSides=4;
    int side1;
    int side2;
    int side3;
    int side4;

    Quadrilateral(int color,int a,int b,int c,int d)
    {
        super(color);
        side1=a;
        side2=b;
        side3=c;
        side4=d;
    }
     void displaySides()
    {
        System.out.println(”Sides are:”+side1+” ”+” ”+side2+” ”+side3+” ”+side4);
    }
}


abstract public class Triangle extends Shape{
    static final int noOfSides=3;
    int side1;
    int side2;
    int side3;
    Triangle(int color,int a,int b,int c)
    {
        super(color);
        side1=a;
        side2=b;
        side3=c;
    }
    void displaySides()
    {
        System.out.println(”Sides are:”+side1+” ”+” ”+side2+” ”+side3);
    }

}
public class Square extends Quadrilateral implements Drawable,Cloneable{
    static final String name=”Square”;
    Square(int color,int a)
    {
        super(color,a,a,a,a);
    }
    void perimeter()
    {
        int perimeter=side1*4;
        System.out.println(”Perimeter is:”+perimeter);
    }
    void area()
    {
        int area=side1*side1;
        System.out.println(”Area is: ”+area);
    }
    void displayName()
    {
        System.out.println(name);
    }
    public void draw()
    {
        new Frame(){
            public void paint(Graphics g)
            {
                if(color==1)
                    g.setColor(Color.red);
                else if(color==2)
                    g.setColor(Color.green);
                else
                    g.setColor(Color.blue);
                g.fillRect(100,100,side1,side2);
            }
            public void setVisible(boolean b)
            {
                super.setVisible(b);
                setSize(500,500);
                addWindowListener(new WindowAdapter() {
                    @Override
                    public void windowClosing(WindowEvent e) {
                        System.exit(0);
                    }
                });
            }
        }.setVisible(true);
    }
}

public class Rectangle extends Quadrilateral implements Drawable,Cloneable{
    static final String name=”Rectangle”;
    Rectangle(int color,int a,int b)
    {
        super(color,a,b,a,b);
    }
    void perimeter()
    {
        int perimeter=2*(side1+side2);
        System.out.println(”Perimeter is: ”+perimeter);
    }
    void area()
    {
        int area=side1*side2;
        System.out.println(”Area is: ”+area);
    }
    void displayName()
    {
        System.out.println(name);
    }
    public void draw()
    {
        new Frame(){
            public void paint(Graphics g)
            {
                if(color==1)
                    g.setColor(Color.red);
                else if(color==2)
                    g.setColor(Color.green);
                else
                    g.setColor(Color.blue);
                g.fillRect(100,100,side1,side2);
            }
            public void setVisible(boolean b)
            {
                super.setVisible(b);
                setSize(500,500);
                addWindowListener(new WindowAdapter() {
                    @Override
                    public void windowClosing(WindowEvent e) {
                        System.exit(0);
                    }
                });
            }
        }.setVisible(true);
    }
}


public class Equilateral extends Triangle{
    static final String name=”Equilateral Triangle”;
    Equilateral(int color,int a)
    {
        super(color,a,a,a);
    }
    void perimeter()
    {
        int perimeter=3*side1;
        System.out.println(”Perimeter is: ”+perimeter);
    }
    void area()
    {
        double area=(Math.sqrt(3)*side1*side2)/4;
        System.out.println(”Area is : ”+area);
    }
    void displayName()
    {
        System.out.println(name);
    }
}


public class RightAngle extends Triangle{
    static final String name="Right Angle Triangle";
    RightAngle(int color,int b,int p,int h)
    {
        super(color,b,p,h);
    }
    void perimeter()
    {
        int perimeter=side1+side2+side3;
        System.out.println("Perimeter is: "+perimeter);
    }
    void area()
    {
        double area=(side1*side2)/2.0;
        System.out.println("Area is: "+area);
    }
    void displayName()
    {
        System.out.println(name);
    }
}


public class Test {
    public static void main(String[] args) {
        Shape[] shapes=new Shape[4];
        shapes[0]=new Square(1,2);
        shapes[1]=new Rectangle(2,2,3);
        shapes[2]=new RightAngle(3,3,4,5);
        shapes[3]=new Equilateral(2,2);

        for(int i=0;i<4;i++)
        {
            shapes[i].displayName();
            shapes[i].displaySides();
            shapes[i].perimeter();
            shapes[i].area();

        }
        Drawable[] d=new Drawable[2];
        d[0]=new Rectangle(1,100,200);
        d[1]=new Square(2,50);
    d[0].draw();
    d[1].draw();

  }
}
```



```
Square
Sides are:2  2 2 2
Perimeter is:8
Area is: 4
Rectangle
Sides are:2  3 2 3
Perimeter is: 10
Area is: 6
Right Angle Triangle
Sides are:3  4 5
Perimeter is: 12
Area is: 6.0
Equilateral Triangle
Sides are:2  2 2
Perimeter is: 6
Area is : 1.7320508075688772
```





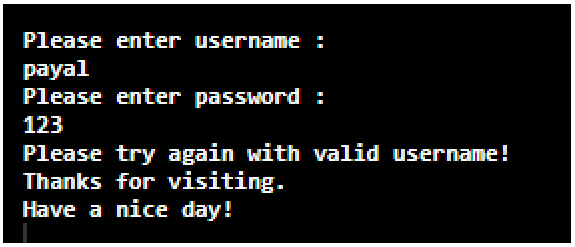
PROGRAM 8

Exception Handling

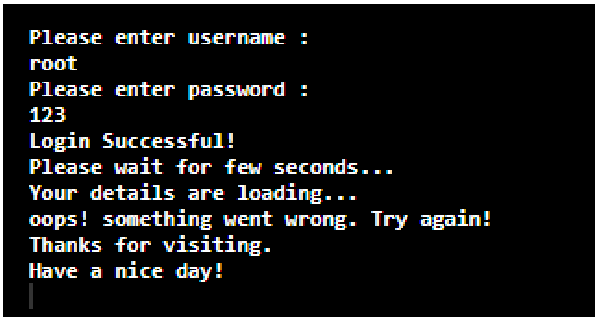
```
import java.util.Scanner;

//custom exception class
class LoginException extends Exception{
    LoginException(String s)
    {
        super(s);
    }
    //override
    public String toString()
    {
        return getMessage();
    }
}

public class ExceptionHandling {
    String details;

    public static void main(String... args)
    {
        Scanner sc=new Scanner(System.in);

        System.out.println(”Please enter username : ”);
        String name=sc.nextLine();
        System.out.println(”Please enter password : ”);
        String pwd=sc.nextLine();

        login(name,pwd);

    }

    public static void login(String name,String pwd)
    {
        ExceptionHandling ex=null;
```

```
try
{
   if(name.equals(”root”))
   {
      if(pwd.equals(”123”))
      System.out.println(”Login Successful!”);
      else
      //if password doesn’t match --->throws custom exception: LoginException
      throw new LoginException(”Please try again with valid password!”);
   }
   //if username doesn’t match --->throws custom exception: LoginException
   else
   throw new LoginException(”Please try again with valid username!”);

   System.out.println(”Please wait for few seconds…”);
   System.out.println(”Your details are loading…”);

   //nested try catch
   try
   {
      Thread.sleep(2000);
   }
   //if thread is interrrupted --->throws InterruptedException
   catch(InterruptedException e)
   {}

   //ex is null --->throws NullPointerException
   System.out.println(ex.details);
}
catch(LoginException e)
{
   System.out.println(e);
}
catch(NullPointerException e)
{
   System.out.println(”oops! something went wrong. Try again!”);
}
finally
{
   System.out.println(”Thanks for visiting.\nHave a nice day!”);
```

```
    }
  }
}
```



```
Please enter username :
payal
Please enter password :
123
Please try again with valid username!
Thanks for visiting.
Have a nice day!
```



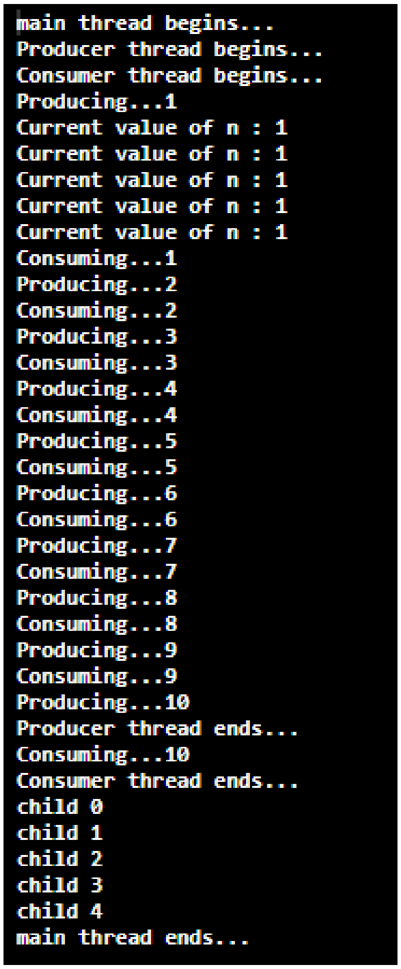
```
Please enter username :
root
Please enter password :
123
Login Successful!
Please wait for few seconds...
Your details are loading...
oops! something went wrong. Try again!
Thanks for visiting.
Have a nice day!
```

PROGRAM 9

Multithreading

```
class Q
{
   int n;
   boolean valueSet=false;

   synchronized int consume()
   {
      while(!valueSet)
      {
         try{wait();}
         catch(InterruptedException e){System.out.println(”Consumer got interrupted”);}
      }
      System.out.println(”Consuming...”+n);
      valueSet=false;
      notify();
      return n;
   }

   synchronized void produce(int n)
   {
      while(valueSet)
      {
         try{wait();}
         catch(InterruptedException e){System.out.println(”Producer got interrupted”);}
      }
      System.out.println(”Producing...”+n);
      this.n=n;
      valueSet=true;
      notify();

   }

   void display()
   {
      for(int i=0;i<5;i++)
      {
         synchronized(this)
         {
            System.out.println(”Current value of n : ”+n);
         }
```

```
            Thread.yield();
        }
    }
}

class Producer implements Runnable{
    Q q;
    Thread t;

    Producer(String name,int p,Q q)
    {
        t=new Thread(this);
        t.setName(name);
        t.setPriority(p);
        this.q=q;
    }

    public void run()
    {
        System.out.println("Producer thread begins...");
        for(int i=1;i<=10;i++)
        {
            q.produce(i);
        }
        System.out.println("Producer thread ends...");
    }
}

class Consumer implements Runnable{
    Q q;
    Thread t;

    Consumer(String name,int p,Q q)
    {
        this.q=q;
        t=new Thread(this);
        t.setName(name);
        t.setPriority(p);
    }

    public void run()
    {
        System.out.println("Consumer thread begins...");
        for(int i=1;i<=10;i++)
        {
            q.consume();
        }
        System.out.println(”Consumer thread ends...”);
    }
}

class Display extends Thread
{
    Q q;

    Display(String name,Q q)
    {
        super(name);
        this.q=q;
    }

    public void run()
    {
        q.display();
    }
}

class NewThread extends Thread
{
    NewThread(String name)
    {
        super(name);
    }
    public void run()
    {
        for(int i=0;i<5;i++)
        {
            System.out.println(getName()+” ”+i);
            try{Thread.sleep(2000);}
            catch(InterruptedException e){}
        }
    }
}

public class MultiThreading
{
    public static void main(String... args) throws InterruptedException
    {
        System.out.println(”main thread begins...”);
        Q q=new Q();
        Producer p=new Producer(”producer”,6,q);
        Consumer c=new Consumer(”consumer”,6,q);
        Display d=new Display(”display”,q);
        NewThread t=new NewThread(”child”);
        p.t.start();
        c.t.start();
        d.start();
        t.start();

        p.t.join();
        c.t.join();
        d.join();
        t.join();
        System.out.println(”main thread ends...”);
    }
}
```

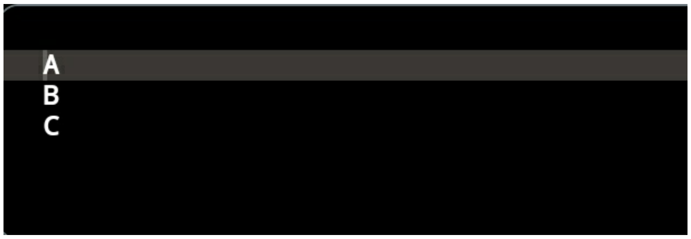


```
main thread begins...
Producer thread begins...
Consumer thread begins...
Producing...1
Current value of n : 1
Current value of n : 1
Current value of n : 1
Current value of n : 1
Current value of n : 1
Consuming...1
Producing...2
Consuming...2
Producing...3
Consuming...3
Producing...4
Consuming...4
Producing...5
Consuming...5
Producing...6
Consuming...6
Producing...7
Consuming...7
Producing...8
Consuming...8
Producing...9
Consuming...9
Producing...10
Producer thread ends...
Consuming...10
Consumer thread ends...
child 0
child 1
child 2
child 3
child 4
main thread ends...
```

PROGRAM 10

Packages

```
//save by A.java
package pack1;
public class A{
    public void msg(){System.out.println("A");}
}
//save by B.java
package pack2;
public class B{
    public void msg(){System.out.println("B");}
}
//save by C.java
package pack3;
public class C{
    public void msg(){System.out.println("C");}
}
//save by D.java
package mypack;
import pack1.*;
import pack2.B;

class B{
    public static void main(String args[]){
        A objA = new A();
        objA.msg();
        B objB = new B();
        objB.msg();
        packC.C objC= new packC.C();
        objC.msg();

  }
}
```



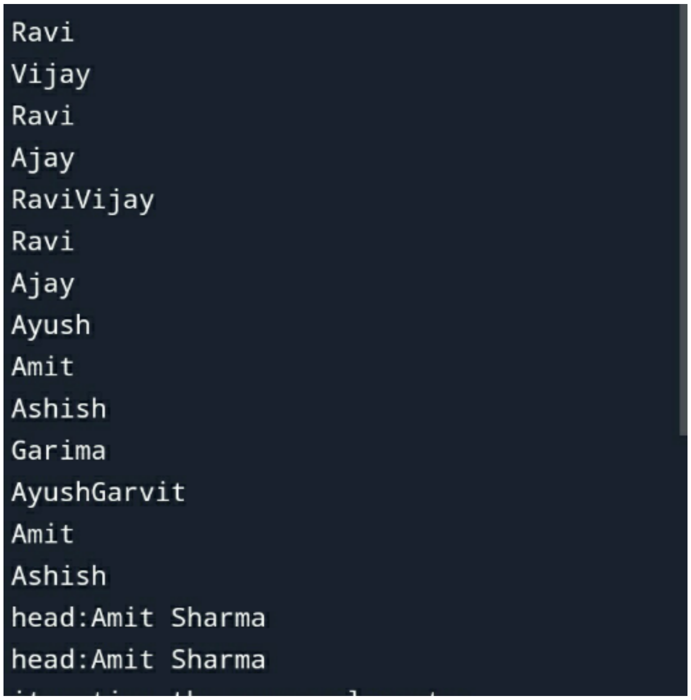
```
A
B
C
```

PROGRAM 11

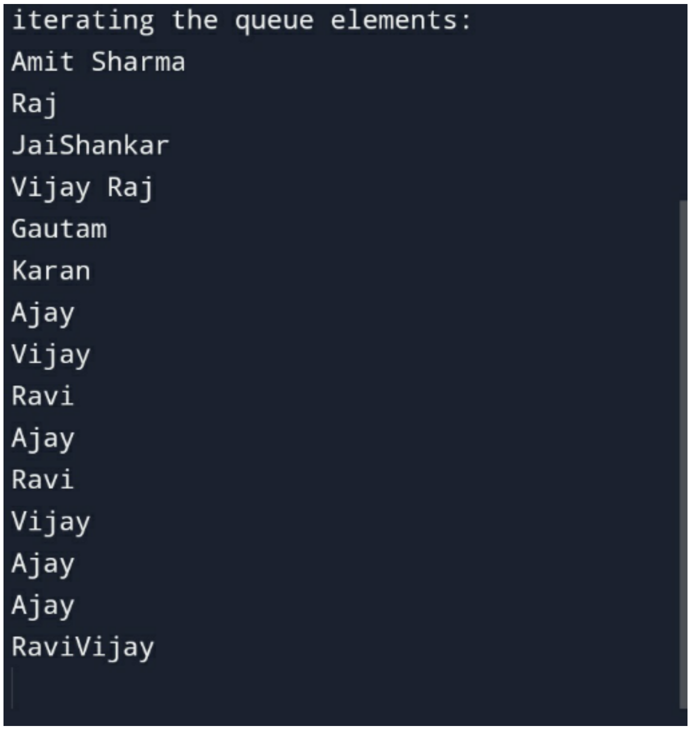
COLLECTION FRAMEWORK

```
import java.util.*;
class TestJavaCollection{
public static void main(String args[]){
        ArrayList<String> list=new ArrayList<String>();//Creating arraylist
        list.add(”Ravi”);//Adding object in arraylist
        list.add(”Vijay”);
        list.add(”Ravi”);
        list.add(”Ajay”);
        //Traversing list through Iterator
        Iterator itr=list.iterator();
        while(itr.hasNext()){
            System.out.println(itr.next());
            }
        LinkedList<String> al=new LinkedList<String>();
        al.add(”Ravi”);
        al.add(”Vijay”);
        al.add(”Ravi”);
        al.add(”Ajay”);
        Iterator<String> itr1=al.iterator();
        while(itr1.hasNext()){
        System.out.println(itr1.next());
        }
        Vector<String> v=new Vector<String>();
        v.add(”Ayush”);
        v.add(”Amit”);
        v.add(”Ashish”);
        v.add(”Garima”);
        Iterator<String> itr2=v.iterator();
        while(itr2.hasNext()){
        System.out.println(itr2.next());
        }
        Stack<String> stack = new Stack<String>();
        stack.push(”Ayush”);
        stack.push(”Garvit”);
        stack.push(”Amit”);
        stack.push(”Ashish”);
        stack.push(”Garima”);
        stack.pop();
        Iterator<String> itr3=stack.iterator();
        while(itr3.hasNext()){
        System.out.println(itr3.next());
        }
        PriorityQueue<String> queue=new PriorityQueue<String>();
        queue.add(”Amit Sharma”);
        queue.add(”Vijay Raj”);
        queue.add(”JaiShankar”);
        queue.add(”Raj”);
        System.out.println(”head:”+queue.element());
        System.out.println(”head:”+queue.peek());
        System.out.println(”iterating the queue elements:”);
        Iterator itr4=queue.iterator();
        while(itr4.hasNext()){
        System.out.println(itr4.next());
        }
        Deque<String> deque = new ArrayDeque<String>();
        deque.add(”Gautam”);
        deque.add(”Karan”);
        deque.add(”Ajay”);
        //Traversing elements
        for (String str : deque) {
        System.out.println(str);
        HashSet<String> set=new HashSet<String>();
        }
        set.add(”Ravi”);
        set.add(”Vijay”);
        set.add(”Ravi”);
        set.add(”Ajay”);
        //Traversing elements
        Iterator<String> itr5=set.iterator();
        while(itr5.hasNext()){
        System.out.println(itr5.next());
```

```
    }
    LinkedHashSet<String> set1=new LinkedHashSet<String>();
    set1.add(”Ravi”);
    set1.add(”Vijay”);
    set1.add(”Ravi”);
    set1.add(”Ajay”);
    Iterator<String> itr6=set1.iterator();
    while(itr6.hasNext()){
    System.out.println(itr6.next());
    }
    TreeSet<String> set2=new TreeSet<String>();
    set2.add(”Ravi”);
    set2.add(”Vijay”);
    set2.add(”Ravi”);
    set2.add(”Ajay”);
    //traversing elements
    Iterator<String> itr7=set2.iterator();
    while(itr7.hasNext()){
    System.out.println(itr7.next());
    }
    }
}
```



```
Ravi
Vijay
Ravi
Ajay
RaviVijay
Ravi
Ajay
Ayush
Amit
Ashish
Garima
AyushGarvit
Amit
Ashish
head:Amit Sharma
head:Amit Sharma
```



```
iterating the queue elements:
Amit Sharma
Raj
JaiShankar
Vijay Raj
Gautam
Karan
Ajay
Vijay
Ravi
Ajay
Ravi
Vijay
Ajay
Ajay
RaviVijay
```

PROGRAM 12
APPLETS

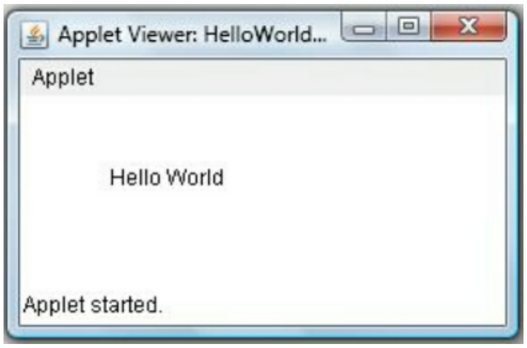
```
//First.java
import java.applet.Applet;
import java.awt.Graphics;
public class First extends Applet{

public void paint(Graphics g){
        g.drawString(”Hello World ”,20,20);
        }
  }
```

Compile the source code to obtain First.class file

```
//MyApplet.html
<html>
<body>
<applet code=”First.class” width=”300” height=”300”>
</applet>
</body>
</html>
```

Following command line will run the applet program :

```
appletviewer MyApplet.html
```



```
Hello World
```

Applet started.

PROGRAM 13

SWING

```
// Java program to create a simple calculator
// with basic +, -, /, * using java swing elements

import java.awt.event.*;
import javax.swing.*;
import java.awt.*;

class calculator extends JFrame implements ActionListener {
static JFrame f;
static JTextField l;
String s0, s1, s2;
 calculator()
{
      s0 = s1 = s2 = ””;
}
 public static void main(String args[])
 {
     f = new JFrame(”calculator”);
     calculator c = new calculator();
        l = new JTextField(16);
        l.setEditable(false);
```

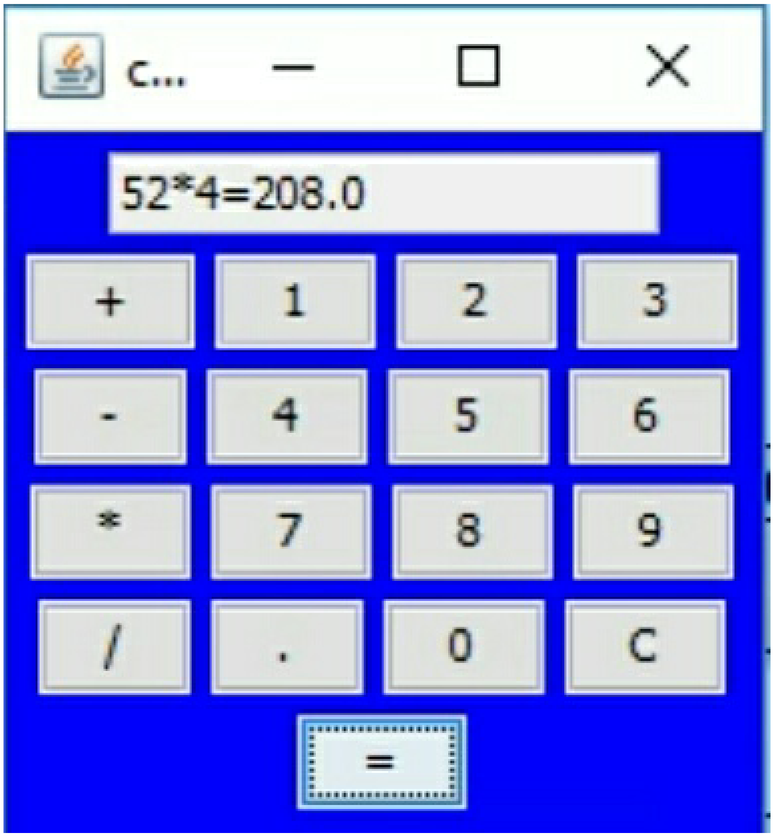
```
f.add(l);
JButton b0, b1, b2, b3, b4, b5, b6, b7, b8, b9, ba, bs, bd, bm, be, beq, beq1;
b0 = new JButton(”0”);
b1 = new JButton(”1”);
b2 = new JButton(”2”);
b3 = new JButton(”3”);
b4 = new JButton(”4”);
b5 = new JButton(”5”);
b6 = new JButton(”6”);
b7 = new JButton(”7”);
b8 = new JButton(”8”);
b9 = new JButton(”9”);
beq1 = new JButton(”=”);
ba = new JButton(”+”);
bs = new JButton(”-”);
bd = new JButton(”/”);
bm = new JButton(”*”);
beq = new JButton(”C”);
be = new JButton(”.”);
JPanel p = new JPanel();
p.setLayoutManager(new GridLayout(4,4,2,2));
bm.addActionListener(c);
bd.addActionListener(c);
bs.addActionListener(c);
ba.addActionListener(c);
```

```
b9.addActionListener(c);
b8.addActionListener(c);
b7.addActionListener(c);
b6.addActionListener(c);
b5.addActionListener(c);
b4.addActionListener(c);
b3.addActionListener(c);
b2.addActionListener(c);
b1.addActionListener(c);
b0.addActionListener(c);
be.addActionListener(c);
beq.addActionListener(c);
beq1.addActionListener(c);
p.add(ba);
p.add(b1);
p.add(b2);
p.add(b3);
p.add(bs);
p.add(b4);
p.add(b5);
p.add(b6);
p.add(bm);
p.add(b7);
p.add(b8);
p.add(b9);
```

```
        p.add(bd);
        p.add(be);
        p.add(b0);
        p.add(beq);
        p.setBackground(Color.blue);
        fadd(p);
        f.add(beq1);
        f.setSize(200,200);
        f.setBackground(Color.blue);
        f.setVisible(true);
}
public void actionPerformed(ActionEvent e)
 {
        String s = e.getActionCommand();
        // if the value is a number
        if ((s.charAt(0) >= ’0’ && s.charAt(0) <= ’9’) || s.charAt(0) == ’.’) {
            // if operand is present then add to second no
            if (!s1.equals(””))
            s2 = s2 + s;
            else
            s0 = s0 + s;
           // set the value of text
           l.setText(s0 + s1 + s2);
        }
        else if (s.charAt(0) == ’C’) {
```

```
        s0 = s1 = s2 = "";
       // set the value of text
       l.setText(s0 + s1 + s2);
    }
    else if (s.charAt(0) == '=') {
       double te;
       if (s1.equals("+"))
          te = (Double.parseDouble(s0) + Double.parseDouble(s2));
       else if (s1.equals("-"))
          te = (Double.parseDouble(s0) - Double.parseDouble(s2));
       else if (s1.equals("/"))
          te = (Double.parseDouble(s0) / Double.parseDouble(s2));
       else
          te = (Double.parseDouble(s0) * Double.parseDouble(s2));
       l.setText(s0 + s1 + s2 + "=" + te);
       s0 = Double.toString(te);
       s1 = s2 = "";
     }
    else {
       if (s1.equals("") || s2.equals(""))
          s1 = s;
       else {
          double te;
          if (s1.equals("+"))
             te = (Double.parseDouble(s0) + Double.parseDouble(s2));
```

```
            else if (s1.equals("-"))
               te = (Double.parseDouble(s0) - Double.parseDouble(s2));
            else if (s1.equals("/"))
               te = (Double.parseDouble(s0) / Double.parseDouble(s2));
            else
               te = (Double.parseDouble(s0) * Double.parseDouble(s2));
            s0 = Double.toString(te);
            s1 = s;
            s2 = "";
            }
        l.setText(s0 + s1 + s2);
        }
    }
}
```


52*4=208.0

PROGRAM 14

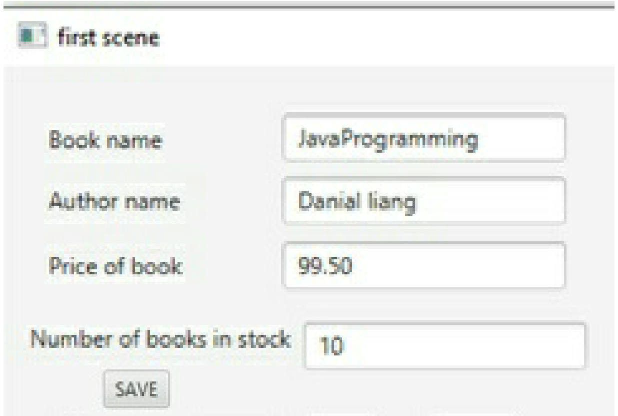
JDBC

```
import javax.swing.*;
import java.awt.event. *;

class MyFrame extends JFrame implements ActionListener
{
   Label  name, author, price, stock;
   JTextField tf1, tf2, tf3, tf4;
   JButton save;

   MyFrame()
   {
      GridBagLayout g=new GridBagLayout();
      GridBagConstraints gc =new GidBagConstraints();
      setLayoutManager(g);
      setTitle(”first scene”);
      Insets i=new Insets(5,5,5,5);
      gc.insets=i;

      name=new JLabel(”Book name  ”);
      author=new JLabel(”Author name ”);
      price=new JLabel(”Price of book ”);
```

```
    stock=new JLabel("Number of books in stock ");
    tf1=new JTextField(15);
    tf2=new JTextField(15);
    tf3=new JTextField(15);
    tf4=new JTextField(15);

    save=new JButton("SAVE");
    save.addActionListener(this);

    gc.gridX=0;
    gc.gridY=0;add(id,gc);
    gc.gridY=1;add(name,gc);
    gc.gridY=2;add(author,gc);
    gc.gridY=3;add(price,gc);
    gc.gridX=1;
    gc.gridY=0;add(tf1,gc);
    gc.gridY=1;add(tf2,gc);
    gc.gridY=2;add(tf3,gc);
    gc.gridY=3;add(tf4,gc);
    gc.gridX=0;gc.gridY=4;
    add(save,gc);

    setBackground(Color.gray);
    setVisible (true );
    setSize(500,500);
}
```

```
 public void actionPerformed(ActionEvent e)
 {
    String nm=tf1.grtText();
    String auth=tf2.getText();
    String pr=tf3.getText();
    String st=tf4.getText();

    Connection con=null;
    try
    {
       Class.forName("com.mysql.jdbc.Driver");

con=DriverManager.getConnection("jdbc:mysql://localhost:3306/book_detalis","root","123");
       PreparedStatement p=con.prepareStatement("Insert into book values(?,?,?,?);");
       p.setString(1,nm);
       p.setString(2,auth);
       p.setInt(3,Integer.parseInt(pr));
       p.setInt(Integer.parseInt(st));
       int r=p.executeUpdate();


    }
    catch(SQLException e)
    {
       System.out.println(e);
```

```
        }
        catch(ClassNotFoundException e)
        {
           System.out.println(e);
        }
        finally
        {
           try {con.close();}
           catch(SQLException e){}
        }
    }
}
```

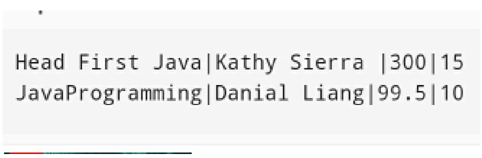


Book name JavaProgramming

Author name Danial liang

Price of book 99.50

Number of books in stock 10



```
Head First Java|Kathy Sierra |300|15
JavaProgramming|Danial Liang|99.5|10
```