A Satyam Roychowdhury Initiative

SNU
SISTER NIVEDITA
UNIVERSITY

SISTER NIVEDITA UNIVERSITY

Computer Programming Practice II

SUBMITTED BY: PIYUSH CHANDRA CHANDRA

DEPT- BTECH (CSE), ROLL- 1027

SUBMITTED TO: Mousumi Bhattacharyya

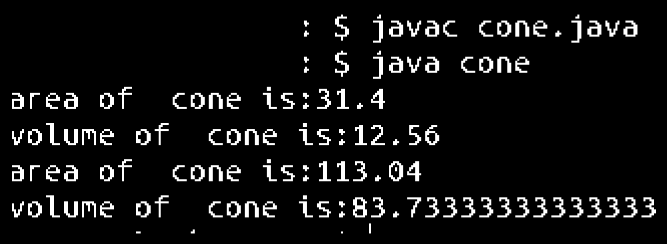
INDEX

1. **Overload a constructor of class cone. Create at least two instances of cone demonstrate constructor overloading. The class has two methods, which return area and volume. Implement "this" keyword in JAVA**

➢
```
import java.lang.*;

class sample

{

        int R,H;

        sample()

        {

                R=H=0;

        }

         sample(int R,int H)

        {

                this.R=R;

                this.H=H;

        }

        void display()

        {

                System.out.println("area of  cone
is:"+((3.14*R*R)+(3.14*R*H)));

                System.out.println("volume of  cone
is:"+(3.14*R*R*H)/3);
```

```
        }

}
class cone
{
        public static void main(String[] args)
        {
                sample s1=new sample(2,3);
                s1.display();
                sample s2=new sample(4,5);
                s2.display();



        }
}
```

OUTPUT:



```
                  : $ javac cone.java
                  : $ java cone
area of  cone is:31.4
volume of  cone is:12.56
area of  cone is:113.04
volume of  cone is:83.73333333333333
```

2. **Create a Complex class to implement complex addition, subtraction, multiplication and division, finding argument of the complex number. Use constructor overloading.**

```
➢ class complex1
  {
      private float real,imag;
      complex1()
      {
          real=400;
          imag=40;
      }
      complex1(int a,int b)
      {
          real=a;
          imag=b;
      }
    public  void  sum(complex1 x,complex1 y)
      {

          this.real=x.real+y.real;
          this.imag=x.imag+y.imag;
          System.out.println("The addition
  is"+this.real+"+"+this.imag+"i");


      }
     public void subtract(complex1 x,complex1 y)
      {

          this.real=x.real-y.real;
          this.imag=x.imag-y.imag;
          System.out.println("The subtraction is"+this.real+"-
  "+this.imag+"i");


      }
    public void multiply(complex1 x,complex1 y)
    {

        this.real=x.real*y.real;
        this.imag=x.imag*y.imag;
        System.out.println("The multiplication
is"+this.real+"*"+this.imag+"i");


    }
    public void divide(complex1 x,complex1 y)
    {

        this.real=x.real/y.real;
        this.imag=x.imag/y.imag;
        System.out.println("The division is
is"+this.real+"/"+this.imag+"i");


    }
}
class complex
{
    public static void main(String args[])
    {
        complex1 c1=new complex1();
        complex1 c2=new complex1(200,20);
        complex1 c3=new complex1();
        c3.sum(c1,c2);
        c3.subtract(c1,c2);
        c3.multiply(c1,c2);
        c3.divide(c1,c2);
    }
}
```

3. **Create a class fraction to implement addition, subtraction, multiplication and division of two proper fractions. This class also contains a private method minimize which can minimizes 2/4 to ½. You have to provide suitable method for taking input and printing the value of the fraction.**

➢
```
 import java.util.*;
public class Fraction
{

   private int numerator, denominator;
 public Fraction()
   {
      numerator = denominator = 0;
   }
 public int getNumerator()
   {
      return numerator;
   }
public void setNumerator(int num)
   {
      numerator=num;
   }
public int getDenominator()
   {
      return denominator;
   }
public void setDenominator(int den)
   {
      denominator=den;
   }
 public Fraction add(Fraction b)
   {

      if ((denominator == 0) || (b.denominator == 0))
         throw new IllegalArgumentException("invalid
denominator");
```

```
        int common = lcd(denominator, b.denominator);

        Fraction commonA = new Fraction();
        Fraction commonB = new Fraction();
        commonA = convert(common);
        commonB = b.convert(common);

        Fraction sum = new Fraction();

        sum.numerator = commonA.numerator +
commonB.numerator;
        sum.denominator = common;

        sum = sum.reduce();
        return sum;
    }
public Fraction subtract(Fraction b)
    {

        if ((denominator == 0) || (b.denominator == 0))
            throw new IllegalArgumentException("invalid
denominator");

        int common = lcd(denominator, b.denominator);

        Fraction commonA = new Fraction();
        Fraction commonB = new Fraction();
        commonA = convert(common);
        commonB = b.convert(common);

        Fraction diff = new Fraction();

        diff.numerator = commonA.numerator -
commonB.numerator;
        diff.denominator = common;
```

```
      diff = diff.reduce();
      return diff;
   }
public Fraction multiply(Fraction b)
   {

      if ((denominator == 0) || (b.denominator == 0))
         throw new IllegalArgumentException("invalid
denominator");

      Fraction product = new Fraction();

      product.numerator = numerator * b.numerator;
      product.denominator = denominator * b.denominator;

      product = product.reduce();
      return product;
   }
 public Fraction divide(Fraction b)
   {

      if ((denominator == 0) || (b.numerator == 0))
         throw new IllegalArgumentException("invalid
denominator");

      Fraction result = new Fraction();

      result.numerator = numerator * b.denominator;
      result.denominator = denominator * b.numerator;

      result = result.reduce();
      return result;
   }
public void input()
   {
   Scanner sc=new Scanner(System.in);
```

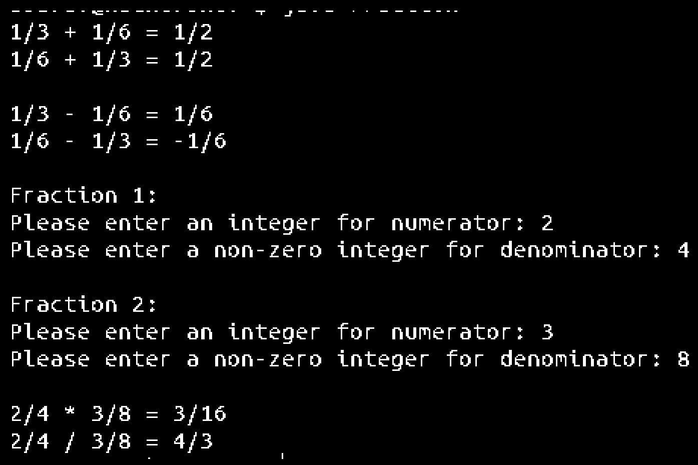
```
      System.out.print("Please enter an integer for numerator: 
");

      numerator = sc.nextInt();

      do {
         System.out.print("Please enter a non-zero integer for 
denominator: ");
         denominator = sc.nextInt();

         if (denominator == 0)
            System.out.println("Invalid value.  Please try again.");
      } while (denominator == 0);
   }
public void output()
   {
       System.out.print(this);
   }
public String toString()
   {
      String buffer = numerator + "/" + denominator;
      return buffer;
   }
private int lcd(int denom1, int denom2)
   {
      int factor = denom1;
      while ((denom1 % denom2) != 0)
         denom1 += factor;
      return denom1;
   }
 private int gcd(int denom1, int denom2)
   {
      int factor = denom2;
      while (denom2 != 0) {
         factor = denom2;
         denom2 = denom1 % denom2;
         denom1 = factor;
```

```
        }
        return denom1;
    }
private Fraction convert(int common)
    {
        Fraction result = new Fraction();
        int factor = common / denominator;
        result.numerator = numerator * factor;
        result.denominator = common;
        return result;
    }
 private Fraction reduce()
    {
        Fraction result = new Fraction();
        int common = 0;

        int num = Math.abs(numerator);
        int den = Math.abs(denominator);

        if (num > den)
           common = gcd(num, den);
        else if (num < den)
           common = gcd(den, num);
        else
           common = num;


        result.numerator = numerator / common;
        result.denominator = denominator / common;
        return result;
    }
 public static void main(String args[])
    {
        Fraction f1 = new Fraction();
        Fraction f2 = new Fraction();

        f1.setNumerator(1);
```

```
f1.setDenominator(3);
f2.setNumerator(1);
f2.setDenominator(6);


Fraction result = new Fraction();

result = f1.add(f2);

System.out.println(f1 + " + " + f2 + " = " + result);

result = f2.add(f1);

System.out.println(f2 + " + " + f1 + " = " + result);
System.out.println();


result = f1.subtract(f2);

System.out.println(f1 + " - " + f2 + " = " + result);

result = f2.subtract(f1);

System.out.println(f2 + " - " + f1 + " = " + result);


System.out.println();
System.out.println("Fraction 1:");
f1.input();
System.out.println();
System.out.println("Fraction 2:");
f2.input();
System.out.println();


result = f1.multiply(f2);
```

```java
        f1.output();
        System.out.print(" * ");
        f2.output();
        System.out.print(" = ");
        result.output();
        System.out.println();


        result = f1.divide(f2);


        f1.output();
        System.out.print(" / ");
        f2.output();
        System.out.print(" = ");
        result.output();
        System.out.println();
    }

}
```

OUTPUT:

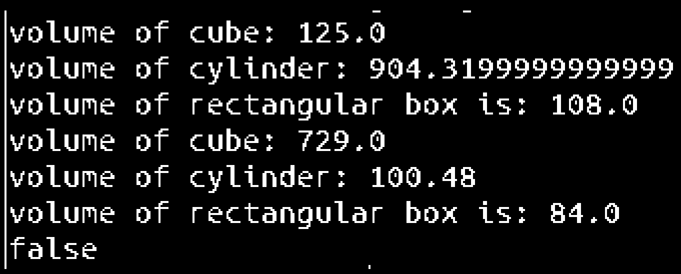


```
1/3 + 1/6 = 1/2
1/6 + 1/3 = 1/2

1/3 - 1/6 = 1/6
1/6 - 1/3 = -1/6

Fraction 1:
Please enter an integer for numerator: 2
Please enter a non-zero integer for denominator: 4

Fraction 2:
Please enter an integer for numerator: 3
Please enter a non-zero integer for denominator: 8

2/4 * 3/8 = 3/16
2/4 / 3/8 = 4/3
```

4. **Define a class volume and then find out the volume of cube, cylinder and rectangular box using method overloading.**

➢
```
class volume
{
        float a,r,h,l,w;
        void vol(float a)
        {
                this.a=a;
                System.out.println("volume of cube: "+(a*a*a));
        }
        void vol( float r,float h)
        {
                this.r=r;
                this.h=h;
                System.out.println("volume of cylinder: "+(3.14*r*r*h));
        }
        void vol(float l,float w,float h)
        {
                this.l=l;
                this.w=w;
                this.h=h;
                System.out.println("volume of rectangular box is: 
"+(l*w*h));
        }
}
class j4
{
        public static void main(String[] args)
        {
                volume v[]=new volume[2];
                v[0]=new volume();
                v[0].vol(5);
                v[0].vol(6,8);
                v[0].vol(6,9,2);

                v[1]=new volume();
                v[1].vol(9);
                v[1].vol(2,8);
                v[1].vol(6,7,2);
                System.out.println(v[0]==v[1]);
        }
}
```

<u>OUTPUT</u>:

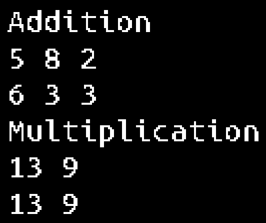


```
volume of cube: 125.0
volume of cylinder: 904.3199999999999
volume of rectangular box is: 108.0
volume of cube: 729.0
volume of cylinder: 100.48
volume of rectangular box is: 84.0
false
```

5. Create a class Matrix to implement addition, multiplication.

➢
```
class matrix
{
        public static void main(String args[])
        {
                int a[][]={{2,3,1},{3,2,1}};
                int b[][]={{3,5,1},{3,1,2}};
                int c[][]=new int[2][3];
                System.out.println("Addition");
                for(int i=0;i<2;i++)
                {
                        for(int j=0;j<3;j++)
                        {
                                c[i][j]=a[i][j]+b[i][j];
                                System.out.print(c[i][j]+" ");
                        }
                        System.out.println();
                }
                System.out.println("Multiplication");
                int d[][]=new int[2][2];
                for(int l=0;l<2;l++)
                {
                        for(int m=0;m<2;m++)
                        {
                                d[l][m]=0;
                                for(int k=0;k<2;k++)
                                {
                                        d[l][m]+=a[l][k]+b[m][k];
                                }
                                System.out.print(d[l][m]+" ");
                        }
                        System.out.println();
                }
        }
}
```

<u>OUTPUT</u>:

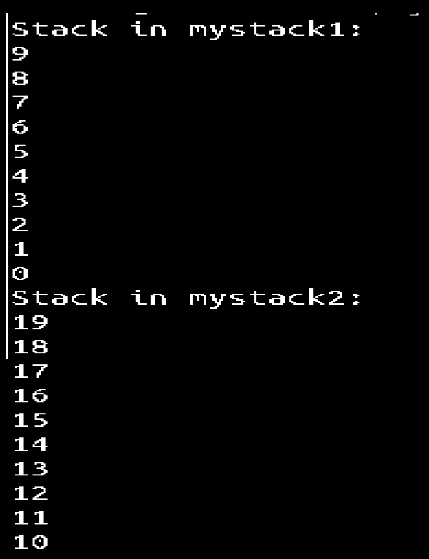


```
Addition
5 8 2
6 3 3
Multiplication
13 9
13 9
```

6. Create class stack and implement suitable methods.

➢
```
class Stack {
int stck[] = new int[10];
int tos;

Stack() {
tos = -1;
}

void push(int item) {
if(tos==9)
System.out.println("Stack is full.");
else
stck[++tos] = item;
}

int pop() {
if(tos < 0) {
System.out.println("Stack underflow.");
return 0;
}
else
return stck[tos--];
}
}

class TestStack {
public static void main(String args[]) {
Stack mystack1 = new Stack();
Stack mystack2 = new Stack();

for(int i=0; i<10; i++) mystack1.push(i);
for(int i=10; i<20; i++) mystack2.push(i);

System.out.println("Stack in mystack1:");
for(int i=0; i<10; i++)
System.out.println(mystack1.pop());
System.out.println("Stack in mystack2:");
for(int i=0; i<10; i++)
System.out.println(mystack2.pop());
 }}
```

OUTPUT:



```
Stack in mystack1:
9
8
7
6
5
4
3
2
1
0
Stack in mystack2:
19
18
17
16
15
14
13
12
11
10
```

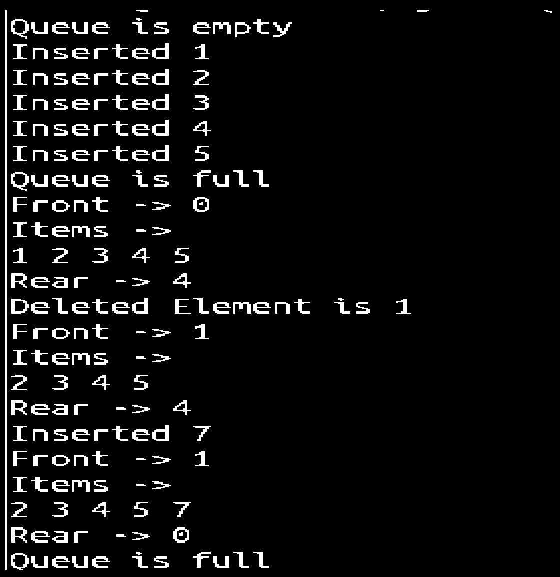
7. **Create class circular queue and implement suitable methods.**

➢ class CQueue {

```
  int SIZE = 5;
  int front, rear;
  int items[] = new int[SIZE];

  CQueue() {
    front = -1;
    rear = -1;
  }


  boolean isFull() {
    if (front == 0 && rear == SIZE - 1) {
      return true;
    }
    if (front == rear + 1) {
      return true;
    }
    return false;
  }
  boolean isEmpty() {
    if (front == -1)
      return true;
    else
      return false;
  }
  void enQueue(int element) {
    if (isFull()) {
      System.out.println("Queue is full");
    } else {
      if (front == -1)
        front = 0;
      rear = (rear + 1) % SIZE;
      items[rear] = element;
      System.out.println("Inserted " + element);
  }
}
int deQueue() {
  int element;
  if (isEmpty()) {
    System.out.println("Queue is empty");
    return (-1);
  } else {
    element = items[front];
    if (front == rear) {
      front = -1;
      rear = -1;
    }
    else {
      front = (front + 1) % SIZE;
    }
    return (element);
  }
}

void display() {

  int i;
  if (isEmpty()) {
    System.out.println("Empty Queue");
  } else {
    System.out.println("Front -> " + front);
    System.out.println("Items -> ");
    for (i = front; i != rear; i = (i + 1) % SIZE)
      System.out.print(items[i] + " ");
    System.out.println(items[i]);
    System.out.println("Rear -> " + rear);
  }
}
public static void main(String[] args) {
  CQueue q = new CQueue();
  q.deQueue();
```

```
    q.enQueue(1);
    q.enQueue(2);
    q.enQueue(3);
    q.enQueue(4);
    q.enQueue(5);
    q.enQueue(6);
    q.display();
    int elem = q.deQueue();
    if (elem != -1) {
      System.out.println("Deleted Element is " + elem);
    }
    q.display();
    q.enQueue(7);
    q.display();
    q.enQueue(8);
  }

}
```

OUTPUT:



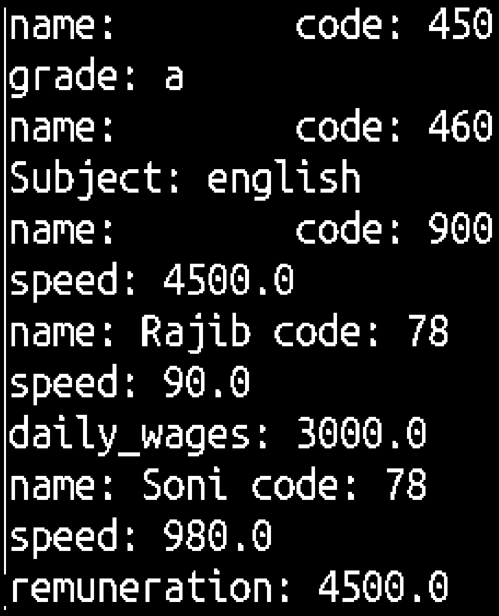
```
Queue is empty
Inserted 1
Inserted 2
Inserted 3
Inserted 4
Inserted 5
Queue is full
Front -> 0
Items ->
1 2 3 4 5
Rear -> 4
Deleted Element is 1
Front -> 1
Items ->
2 3 4 5
Rear -> 4
Inserted 7
Front -> 1
Items ->
2 3 4 5 7
Rear -> 0
Queue is full
```

8. **Write a java program using the following class hierarchy of an educational institution. Class Staff (name, code). Class Officer (grade), subclass of Staff. Class Teacher (subject), subclass of Staff.Class Typist (speed), subclass of Staff.Class CasualTypist (daily wages), subclass Typist.Class RegularTypist (remuneration), subclass of Typist.Create different objects of each class except Staff, input values for different variables for each one, store these objects using array.**

➢
```
importjava.util.*;
class staff
{
   String name;
int code;
staff(String n,int c)
   {
        name=n;
        code=c;
   }
void display()
   {
System.out.println("Name:" +name);
System.out.println("Code:"+code);
   }
}
class officer extends staff
{
   String grade;
officer(String name,intcode,String g)
   {
        super(name,code);
grade=g;
   }
void display()
   {
super.display();
System.out.println("Grade:"+grade);
   }
}
class teacher extends staff{
String subject;
teacher(String name,intcode,String s)
{
super(name,code);
subject=s;
}
void display()
{
super.display();
System.out.println("Subject:"+subject);
}
}
class typist extends staff
{
int speed;
typist(String name,intcode,intsp)
   {
        super(name,code);
speed=sp;
   }
void display()
   {
super.display();
System.out.println("Speed:"+speed);
   }
}
classcasualtypist extends typist{
intdailywage;
casualtypist(String name,intcode,intspeed,intdw)
{
super(name,code,speed);
dailywage=dw;
}
```

```
void display()
{
super.display();
System.out.println("Daily Wage:"+dailywage);
}
}
classregulartypist extends typist{
intrenu;
regulartypist(String name,intcode,intspeed,int r)
{
super(name,code,speed);
renu=r;
}
void display()
{
super.display();
System.out.println("Renumeration:"+renu);
}
}
public class program8
{
public static void main(String args[])
  {
        Scanner sc=new Scanner(System.in);
      int code;
        String name;
      System.out.println("Enter the name and code");
      name=sc.nextLine();
      code=sc.nextInt();
staff a[]=new staff[20];
for(inti=0;i<2;i++)
     {
      a[i]=new staff(name,code);
     }
officer f= new officer("Sarit",code,"A");
teacher t=new teacher("",code,"Mathematics");
typist t1=new typist("",code,30);
casualtypist t2=new casualtypist("",code,30,3000);
regulartypist t3=new regulartypist("",code,30,1000);
f.display();
   }
}
```

OUTPUT:



```
name:        code: 450
grade: a
name:        code: 460
Subject: english
name:        code: 900
speed: 4500.0
name: Rajib code: 78
speed: 90.0
daily_wages: 3000.0
name: Soni code: 78
speed: 980.0
remuneration: 4500.0
```

9. **Assume that a bank maintains two kinds of accounts for their customer: first one is called savings account and the other is called current account. The savings account holders should also maintain a minimum balance and if the balance falls below this level, a service charge imposed. Create a class account that stores customer name, account number and type of account. From this, derive classes CrrAcc and SavAcc to make them more specific to their requirements. Include the necessary methods to achieve the following tasks: Accept deposit from a customer and update the balance.Display the balance.Compute and deposit interest.Permit withdrawal and update the balance.Check for minimum balance, impose penalty, if necessary, and update the balance.**

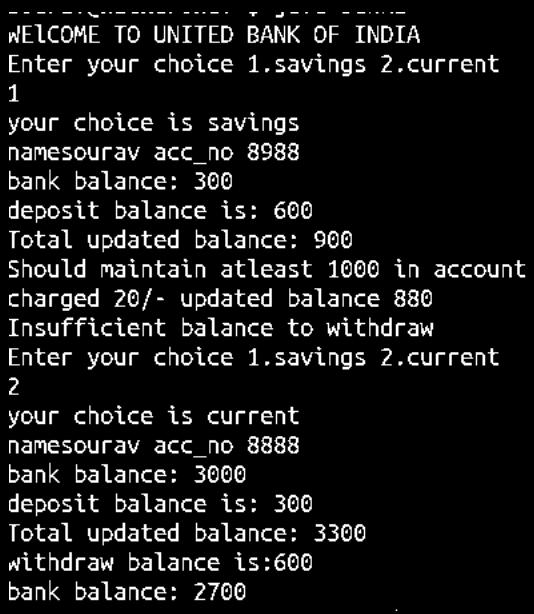
➢
```
import java.util.Scanner;
class account
{
    protected String custName;
    protected int accNumber;
    private String typeOfAccount;
    account()
    {
        this.custName="NULL";
        this.accNumber=0000;
        this.typeOfAccount="NULL";

    }
    account(String cname,int anumber)
    {
        this.custName=cname;
        this.accNumber=anumber;


    }

}
```

```
class savingsAccount extends account{
    private int totalBalance;
   private int amount;
    int mimBalance=1000;

    savingsAccount(account acc,int depositFirstAmmount)
    {
            this.custName=acc.custName;
            this.accNumber=acc.accNumber;
            this.totalBalance=depositFirstAmmount;

    }
void deposit()
{
    System.out.println("Enter the ammount you want to deposit");
    Scanner sc=new Scanner(System.in);
    amount=sc.nextInt();
    totalBalance=totalBalance+amount;

}
        void checkMimBalance()
        {
            if(totalBalance<mimBalance)
            {
                totalBalance=totalBalance-50;
            }
        }
void withdrawl()
{
    System.out.println("Enter the ammount you want to withdraw");
    Scanner sc=new Scanner(System.in);
    amount=sc.nextInt();
    if(amount>totalBalance){
        System.out.println("You donot have that much amount to
withdraw");
    }
    else{
```

```
    totalBalance=totalBalance-amount;
    }

}
void showBalance(){
    checkMimBalance();
    System.out.println("The total balance in your account is:
"+totalBalance);

}

}
class currentAccount extends account
{
    private int totalBalance;
   private int amount;

    protected int currentTotalBalance;

    currentAccount(account acc)
    {
            this.custName=acc.custName;
            this.accNumber=acc.accNumber;
    }
    void deposit()
    {

    System.out.println("Enter the ammount you want to deposit");
    Scanner sc=new Scanner(System.in);
    amount=sc.nextInt();
    totalBalance=totalBalance+amount;
    }
    void withdrawl()
    {

    System.out.println("Enter the ammount you want to withdraw");
    Scanner sc=new Scanner(System.in);
    amount=sc.nextInt();
    if(amount>totalBalance){
        System.out.println("You donot have that much amount to
withdraw");
    }
    else{
    totalBalance=totalBalance-amount;

    }
}
    void showBalance(){
        System.out.println("The total balance in your account is:
"+totalBalance);

    }
}


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

            account acc=new account("Ranson",742123);
            savingsAccount sa=new savingsAccount(acc,2500);
            currentAccount ca=new currentAccount(acc);
            System.out.println("**** Welecome to banking ****");
            int toa,flag=1;
            while(flag!=0)
            {

            System.out.println("Enter type of account you want to
perform operations 1.Savings Account 2.Current Account 3.Exit");
            toa=sc.nextInt();

            switch(toa)
            {
                case 1:{
                  int ch=0;
                  boolean valid=true;

                  while(valid!=false)
                  {
                  System.out.println("Enter the operation you want to
perform "+" 1.Deposit 2.Withdrwal 3.Display Balance 4.Back");

                  ch=sc.nextInt();

                        switch(ch)
                        {
                           case 1: sa.deposit();
                              break;
                           case 2:sa.withdrawl();
                           break;
                           case 3:sa.showBalance();
                           break;
                           case 4: valid=false;
                              break;
                           default: System.out.println("Please enter a
valid choice");
                        }
                  }
                  break;
                }
                case 2: {
                  int ch;
                     int valid=1;
                  while(valid!=0)
                  {
                  System.out.println("Enter the operation you want to
perform "+" 1.Deposit 2.Withdrwal 3.Display Balance 4.Back");

                  ch=sc.nextInt();
```

```
                    switch(ch)
                    {
                       case 1: ca.deposit();
                          break;
                       case 2:ca.withdrawl();
                       break;
                       case 3:ca.showBalance();
                       break;
                       case 4:valid=0;
                       break;
                       default: System.out.println("Please enter a
valid choice");
                    }
              }
             break;
            }
            case 3:flag=0;
            break;
            default:System.out.println("You have entered a wrong
choice");




      }
   }
}
}
```

OUTPUT:



```
WELCOME TO UNITED BANK OF INDIA
Enter your choice 1.savings 2.current
1
your choice is savings
namesourav acc_no 8988
bank balance: 300
deposit balance is: 600
Total updated balance: 900
Should maintain atleast 1000 in account
charged 20/- updated balance 880
Insufficient balance to withdraw
Enter your choice 1.savings 2.current
2
your choice is current
namesourav acc_no 8888
bank balance: 3000
deposit balance is: 300
Total updated balance: 3300
withdraw balance is:600
bank balance: 2700
```

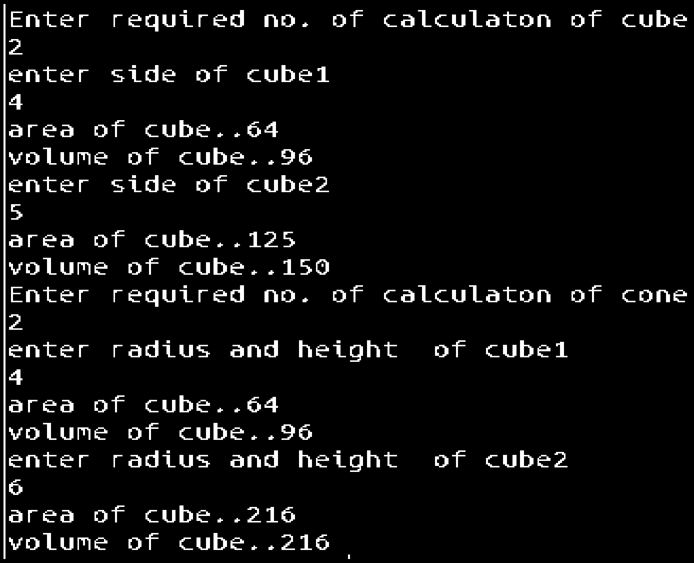
10. Create an abstract class called shape to model a three dimensional shape. Derive classes cube and cone. Let user specify the size. The user may specify many items of the same shape. All the instances of each shape should be stored into a single array. Write a program to calculate area and volume of each shape.

➢
```
import java.lang.Math;
import java.util.Scanner;
abstract class shape
{

    abstract public double area();
    abstract public double volume();
    abstract public void getdata();

}
class cube extends shape
{

    int a;


   public void getdata()
    {
        Scanner sc=new Scanner(System.in);
        float ar;
        System.out.println("Enter side of cube:");
        ar=sc.nextInt();


    }
    public double volume()
```

```
        {
            float vol;
            Scanner sc=new Scanner(System.in);

            vol=a*a*a;
            System.out.println("The volume of cube is:"+vol);
            return vol;

        }
        public double area()
        {
            float ar;
           ar=6*a*a;
           System.out.println("The area of cube is"+ar);
           return ar;
        }
    }
    class cone extends shape
    {
        Scanner sc=new Scanner(System.in);
        int r,h;
        double vol;
        double ar;
       public void getdata()
        {
            System.out.println("Enter radius and height:");
            r=sc.nextInt();
            h=sc.nextInt();

        }
        public  double volume()
        {
```

```
        vol=(3.14*r*r*h)/3;
        System.out.println("The volume of cone is:"+vol);
        return vol;
    }
    public double area()
    {
        ar=3.14*r*(r+Math.sqrt(r*r+h*h));
        System.out.println("The area of cone is:"+ar);
        return ar;

    }
}
class abs
{
    public static void main(String args[])
    {
      shape s[]=new shape[2];

      s[0]=new cone();
      s[1]=new cube();
      s[0].getdata();
      s[1].getdata();

      s[0].volume();

      s[1].volume();
      s[0].area();
      s[1].area();
}
}
```

OUTPUT:

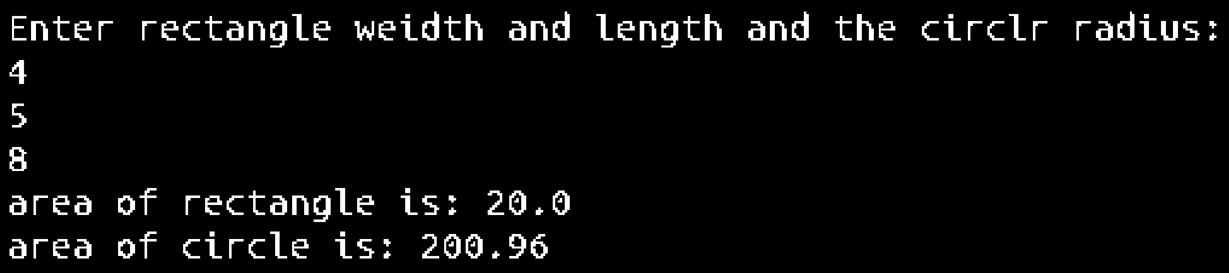


```
Enter required no. of calculaton of cube
2
enter side of cube1
4
area of cube..64
volume of cube..96
enter side of cube2
5
area of cube..125
volume of cube..150
Enter required no. of calculaton of cone
2
enter radius and height  of cube1
4
area of cube..64
volume of cube..96
enter radius and height  of cube2
6
area of cube..216
volume of cube..216
```

11. Using interface calculate the area of the rectangle and circle.

➢
```
import java.util.Scanner;
interface sample
{
         void area1();
         void area2();
}

class shape implements sample
{       Scanner Sc=new Scanner(System.in);
        float w=Sc.nextFloat();
        float l=Sc.nextFloat();
        float r=Sc.nextFloat();
        double pi=3.14;
        @Override
         public void area1()
        {

                System.out.println("area of rectangle is: "+(w*l));
        }

        @Override
        public void area2()
        {

                System.out.println("area of circle is: "+(pi*r*r));
        }



}


class j100
{
        public static void main(String args[])
        {       System.out.println("Enter rectangle weidth and
    length and the circlr radius: ");
                sample s=new shape();
                s.area1();

                s.area2();
        }
```

OUTPUT:

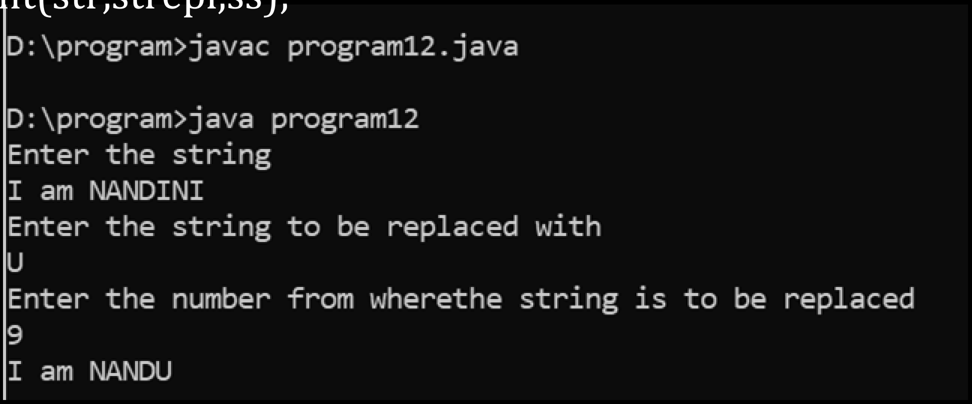


```
Enter rectangle weidth and length and the circlr radius:
4
5
8
area of rectangle is: 20.0
area of circle is: 200.96
```

12. Replace a particular substring from a given string.

```
➢ importjava.util.*;
importjava.util.*;
class program
{
int l;
void replacement(String str,Stringstrepl,intss)
{
String s2,s ;
s=str.substring(ss);
System.out.println(str.replace(s,strepl));
}
}
public class program12
{
public static void main(String args[])
{
program p=new program();
Scanner sc=new Scanner (System.in);
String str,strepl;
intss;
System.out.println("Enter the string");
str=sc.nextLine();
System.out.println("Enter the string to be replaced with");
strepl=sc.nextLine();
System.out.println("Enter the number from wherethe string
is to be replaced");
ss=sc.nextInt();
p.replacement(str,strepl,ss);
}
}
```

OUTPUT:

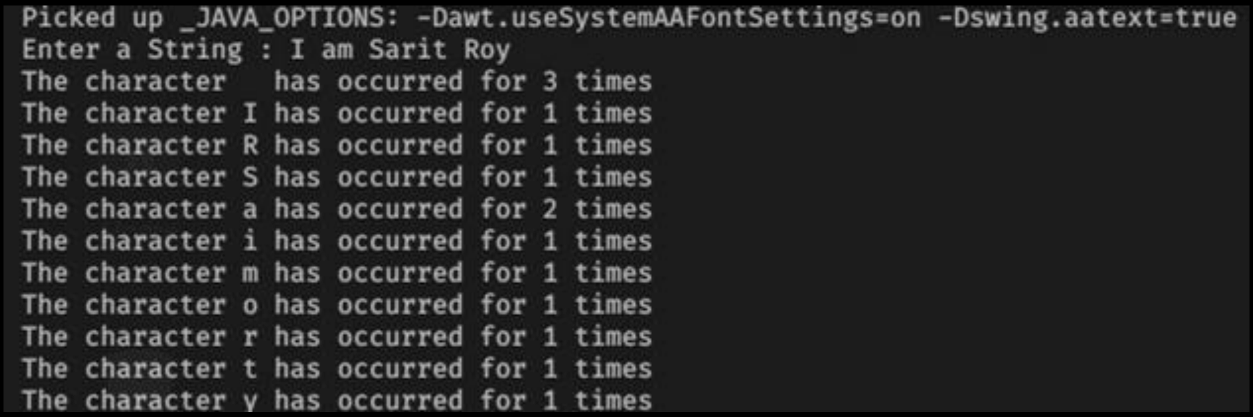


```
D:\program>javac program12.java

D:\program>java program12
Enter the string
I am NANDINI
Enter the string to be replaced with
U
Enter the number from wherethe string is to be replaced
9
I am NANDU
```

13. Calculate the frequency of each character of a given string.

➢
```
importjava.util.*;
class program
{
finalint SIZE = 256;
        inti;
void frequency(String str)
   {
int n = str.length();

intfreq[] = new int[SIZE];
for (i = 0; i< n; i++) {
freq[(int) str.charAt(i)]++;
       }
       // Print Frequency of characters
for (i = 0; i< SIZE; i++) {
if (freq[i] != 0) {
System.out.println("The character " + (char) i  + " has occurred for " + freq[i] + " times");
          }
       }
   }
       }
public class program13
        {
public static void main(String args[])
   {
                program p=new program();
       String str;
       Scanner sc = new Scanner(System.in);
str = sc.nextLine();
p.frequency(str);
   }
}
```

OUTPUT:



```
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Enter a String : I am Sarit Roy
The character   has occurred for 3 times
The character I has occurred for 1 times
The character R has occurred for 1 times
The character S has occurred for 1 times
The character a has occurred for 2 times
The character i has occurred for 1 times
The character m has occurred for 1 times
The character o has occurred for 1 times
The character r has occurred for 1 times
The character t has occurred for 1 times
The character y has occurred for 1 times
```

14. Copy a string from one file to another without terminating character.

➢
```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class CopyExample
{
   public static void main(String[] args)
   {
        FileInputStream instream = null;
        FileOutputStream outstream = null;

        try{
            File infile =new File("C:\\MyInputFile.txt");
            File outfile =new File("C:\\MyOutputFile.txt");

            instream = new FileInputStream(infile);
            outstream = new FileOutputStream(outfile);

            byte[] buffer = new byte[1024];

            int length;
            /*copying the contents from input stream to
             * output stream using read and write methods
             */
            while ((length = instream.read(buffer)) > 0){
                 outstream.write(buffer, 0, length);
            }

            //Closing the input/output file streams
            instream.close();
            outstream.close();

            System.out.println("File copied successfully!!");
```

```
                }catch(IOException ioe){
                        ioe.printStackTrace();
                 }
            }
        }
```

OUTPUT:

```
File copied successfully!!
```

15. Count frequency of each character in a file.

```
➢   public class Frequency
  {
      public static void main(String[] args) {
          String str = "picture perfect";
          int[] freq = new int[str.length()];
          int i, j;

          //Converts given string into character array
          char string[] = str.toCharArray();

          for(i = 0; i <str.length(); i++) {
              freq[i] = 1;
              for(j = i+1; j <str.length(); j++) {
                  if(string[i] == string[j]) {
                      freq[i]++;

                      //Set string[j] to 0 to avoid printing visited
  character
                      string[j] = '0';
                  }
              }
          }

          //Displays the each character and their
  corresponding frequency
          System.out.println("Characters and their
  corresponding frequencies");
          for(i = 0; i <freq.length; i++) {
              if(string[i] != ' ' && string[i] != '0')
                  System.out.println(string[i] + "-" + freq[i]);
          }
      }
  }
```

OUTPUT:

```
Characters and their corresponding frequencies
p-2
i-1
c-2
t-2
u-1
r-2
e-3
f-1
```

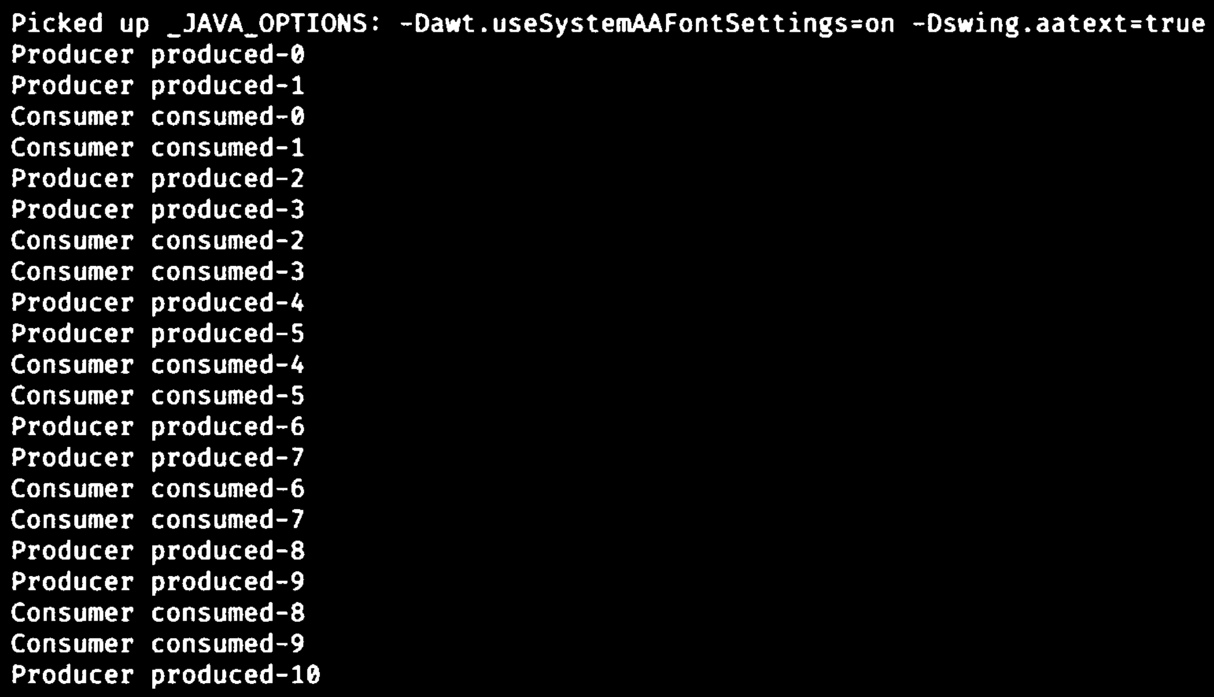
16. Copy the context from one file to another file located to another machine (use socket).

➢
```
import java.util.Scanner;
interface sample
{
         void area1();
         void area2();
}
class shape implements sample
{       Scanner Sc=new Scanner(System.in);
        float w=Sc.nextFloat();
        float l=Sc.nextFloat();
        float r=Sc.nextFloat();
        double pi=3.14;
        @Override
         public void area1()
        {

                System.out.println("area of rectangle is: "+(w*l));
        }
        @Override
        public void area2()
        {

                System.out.println("area of circle is: "+(pi*r*r));
        }
}
class j100
{
        public static void main(String args[])
        {       System.out.println("Enter rectangle weidth and
length and the circlr radius: ");
                sample s=new shape();
                s.area1();
                s.area2();
        }
```

17. Implement Bounded Buffer Problem.

➢
```
import java.util.LinkedList;
public class Threadexample {
public static void main(String[] args)
throws InterruptedException
{
final PC pc = new PC();
Thread t1 = new Thread(new Runnable() {
@Override
public void run()
{
try {
pc.produce();
}
catch (InterruptedException e) {
e.printStackTrace();
}
}
});
Thread t2 = new Thread(new Runnable() {
@Override
public void run()
{
try {
pc.consume();
}
catch (InterruptedException e) {
e.printStackTrace();
}
}
});
t1.start();
t2.start();
t1.join();
t2.join();
}
public static class PC {
LinkedList<Integer> list = new LinkedList<>();
44 | P a g e b y S a r i t R o y
int capacity = 2;
public void produce() throws InterruptedException
{
int value = 0;
while (true) {
synchronized (this)
{
while (list.size() == capacity)
wait();
System.out.println("Producer produced-"
+ value);
list.add(value++);
notify();
Thread.sleep(1000);
}
}
}
public void consume() throws InterruptedException
{
while (true) {
synchronized (this)
{
while (list.size() == 0)
wait();
int val = list.removeFirst();
System.out.println("Consumer consumed-"
+ val);
notify();
Thread.sleep(1000);
}
}
}
}
}
```

OUTPUT:



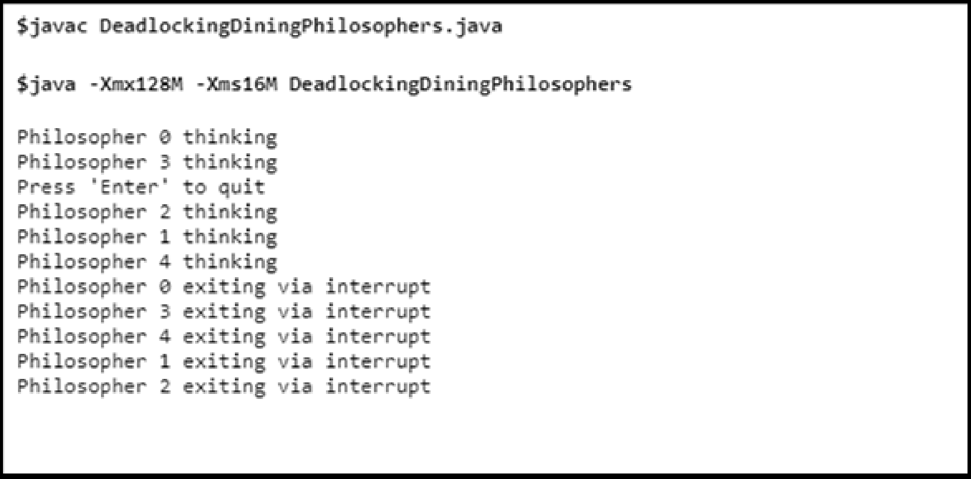
```
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Producer produced-0
Producer produced-1
Consumer consumed-0
Consumer consumed-1
Producer produced-2
Producer produced-3
Consumer consumed-2
Consumer consumed-3
Producer produced-4
Producer produced-5
Consumer consumed-4
Consumer consumed-5
Producer produced-6
Producer produced-7
Consumer consumed-6
Consumer consumed-7
Producer produced-8
Producer produced-9
Consumer consumed-8
Consumer consumed-9
Producer produced-10
```

18. Implement Dining Philosophers Problem (create deadlock).

➢
```
import java.util.concurrent.*;
import java.util.*;

class Chopstick {

private boolean taken = false;

public synchronized
void take() throws InterruptedException {
while (taken) {
wait();
}
taken = true;
}

public synchronized void drop() {
taken = false;
notifyAll();
}
}

class Philosopher implements Runnable {

private Chopstick left;
private Chopstick right;
private final int id;
private final int ponderFactor;
private Random rand = new Random(47);

private void pause() throws InterruptedException {
if (ponderFactor == 0) {
return;
}
TimeUnit.MILLISECONDS.sleep(
rand.nextInt(ponderFactor * 250));
}

public Philosopher(Chopstick left, Chopstick right,
int ident, int ponder) {
this.left = left;
this.right = right;
id = ident;
ponderFactor = ponder;
}

public void run() {
try {
while (!Thread.interrupted()) {
System.out.println(this + " " + "thinking");
pause();
// Philosopher becomes hungry
System.out.println(this + " " + "grabbing right");
right.take();
System.out.println(this + " " + "grabbing left");
left.take();
System.out.println(this + " " + "eating");
pause();
right.drop();
left.drop();
}
} catch (InterruptedException e) {
System.out.println(this + " " + "exiting via interrupt");
}
}

public String toString() {
return "Philosopher " + id;
}
}

public class DeadlockingDiningPhilosophers {
```

```
public static void main(String[] args) throws Exception {
int ponder = 5;
if (args.length > 0) {
ponder = Integer.parseInt(args[0]);
}
int size = 5;
if (args.length > 1) {
size = Integer.parseInt(args[1]);
}
ExecutorService exec = Executors.newCachedThreadPool();
Chopstick[] sticks = new Chopstick[size];
for (int i = 0; i < size; i++) {
sticks[i] = new Chopstick();
}
for (int i = 0; i < size; i++) {
exec.execute(new Philosopher(
sticks[i], sticks[(i + 1) % size], i, ponder));
}
if (args.length == 3 && args[2].equals("timeout")) {
TimeUnit.SECONDS.sleep(5);
} else {
System.out.println("Press 'Enter' to quit");
System.in.read();
}
exec.shutdownNow();
}
}
```

OUTPUT:



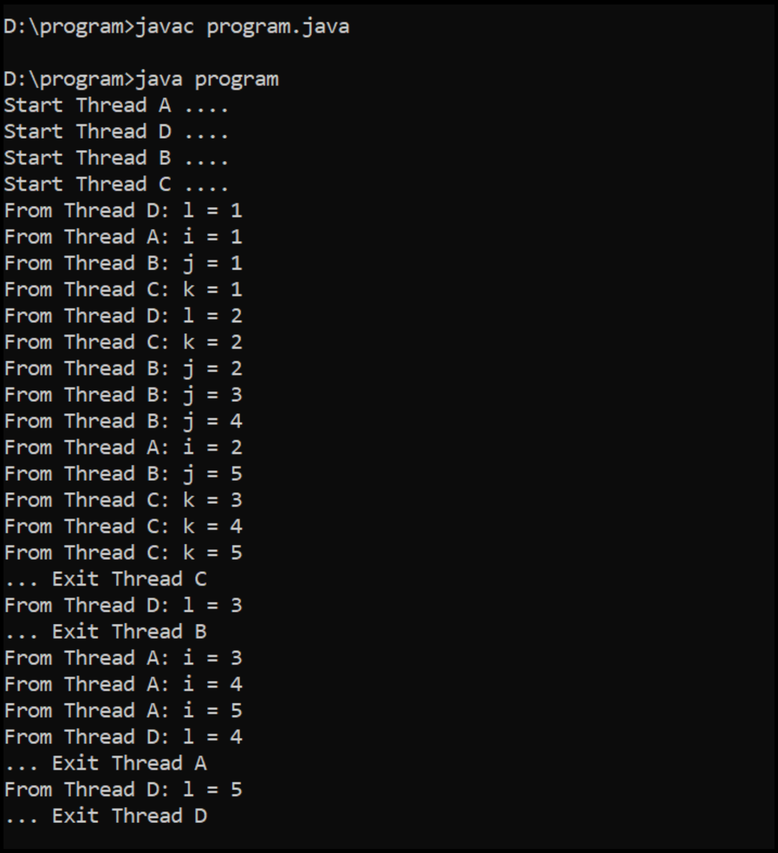
```
$javac DeadlockingDiningPhilosophers.java

$java -Xmx128M -Xms16M DeadlockingDiningPhilosophers

Philosopher 0 thinking
Philosopher 3 thinking
Press 'Enter' to quit
Philosopher 2 thinking
Philosopher 1 thinking
Philosopher 4 thinking
Philosopher 0 exiting via interrupt
Philosopher 3 exiting via interrupt
Philosopher 4 exiting via interrupt
Philosopher 1 exiting via interrupt
Philosopher 2 exiting via interrupt
```

19. Create four threads with priority 1,3,5,7 respectively. Update a counter in each of the threads for ten sec. print the final value of the count from each thread.

```
➢ classClassA extends Thread{
          public void run() {
                    System.out.println("Start Thread A ....");
                    for(inti = 1; i<= 5; i++) {
                              System.out.println("From Thread A: i = "+ i);
                    }
                    System.out.println("... Exit Thread A");
          }
  }

  classClassB extends Thread{
  public void run() {
          System.out.println("Start Thread B ....");
          for(int j = 1; j <= 5; j++) {
                    System.out.println("From Thread B: j = "+ j);
          }
          System.out.println("... Exit Thread B");
          }
  }

  classClassC extends Thread{
          public void run() {
          System.out.println("Start Thread C ....");
          for(int k = 1; k <= 5; k++) {
                    System.out.println("From Thread C: k = "+ k);
          }
          System.out.println("... Exit Thread C" );
          }
  }
  classClassD extends Thread{
          public void run() {
          System.out.println("Start Thread D ....");
```

```
        for(int k = 1; k <= 5; k++) {
                System.out.println("From Thread D: l = "+ k);
        }
        System.out.println("... Exit Thread D" );
        }
}


public class program{
        public static void main (String args[]) {
                ClassA t1 = new ClassA();
                ClassB t2 = new ClassB();
                ClassC t3 = new ClassC();
ClassD t4=new ClassD();
                t3.setPriority(10);
                t2.setPriority(Thread.NORM_PRIORITY);
                t1.setPriority(Thread.MIN_PRIORITY);
                t4.setPriority(7);

                t1.start(); t2.start(); t3.start(); t4.start();

        }
}
```

<u>OUTPUT</u>:

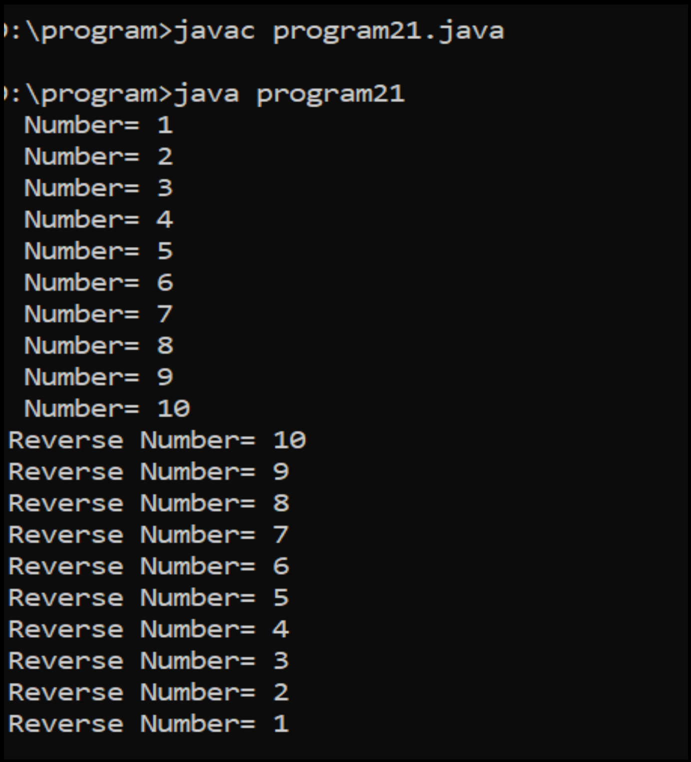


```
D:\program>javac program.java

D:\program>java program
Start Thread A ....
Start Thread D ....
Start Thread B ....
Start Thread C ....
From Thread D: l = 1
From Thread A: i = 1
From Thread B: j = 1
From Thread C: k = 1
From Thread D: l = 2
From Thread C: k = 2
From Thread B: j = 2
From Thread B: j = 3
From Thread B: j = 4
From Thread A: i = 2
From Thread B: j = 5
From Thread C: k = 3
From Thread C: k = 4
From Thread C: k = 5
... Exit Thread C
From Thread D: l = 3
... Exit Thread B
From Thread A: i = 3
From Thread A: i = 4
From Thread A: i = 5
From Thread D: l = 4
... Exit Thread A
From Thread D: l = 5
... Exit Thread D
```

20. Create two threads; one will print from 1 to 10. Another will print 10 to 1. In the second thread if value is 6 it will sleep for 10000 milliseconds.

➢
```
importjava.lang.*;
import java.io.*;
importjava.util.*;
class number extends Thread
{
        public synchronized void run()
        {
                try
                {
                        for(inti=1;i<=10;i++)
                        {

                                        System.out.println ("  Number= 
"+i);
                                }
                                Thread.sleep(1000);
                        }

                catch (Exception e){}
        }
}
        classrev_number extends Thread
{
        public synchronized void run()
        {
                try
                {
                        for(inti=10;i>=1;i--)
                        {

                                        System.out.println (" Reverse 
Number= "+i);
                                        if(i==6)
                                        {
                                        Thread.sleep(10000);
                                        }
                                }

                                Thread.sleep(1000);
                        }

                catch (Exception e){}
        }
}
        public class program21
{
        public static void main(String args[])
        {
                number n1=new number();
                try
                {n1.start();
                n1.join();
                }
                catch(InterruptedException e){}
                newrev_number().start();
}
}
```

OUTPUT:



```
):\program>javac program21.java

):\program>java program21
 Number= 1
 Number= 2
 Number= 3
 Number= 4
 Number= 5
 Number= 6
 Number= 7
 Number= 8
 Number= 9
 Number= 10
Reverse Number= 10
Reverse Number= 9
Reverse Number= 8
Reverse Number= 7
Reverse Number= 6
Reverse Number= 5
Reverse Number= 4
Reverse Number= 3
Reverse Number= 2
Reverse Number= 1
```

21. Write an applet program in java to design a calculator.

➢
```
import java.awt.*;
import java.awt.event.*;

public class MyCalculator extends Frame
{

public boolean setClear=true;
double number, memValue;
char op;

String digitButtonText[] = {"7", "8", "9", "4", "5", "6",
"1", "2", "3", "0", "+/-", "." };
String operatorButtonText[] = {"/", "sqrt", "*", "%", "-",
"1/X", "+", "=" };
String memoryButtonText[] = {"MC", "MR", "MS", "M+"
};
String specialButtonText[] = {"Backspc", "C", "CE" };

MyDigitButton digitButton[]=new
MyDigitButton[digitButtonText.length];
MyOperatorButton operatorButton[]=new
MyOperatorButton[operatorButtonText.length];
MyMemoryButton memoryButton[]=new
MyMemoryButton[memoryButtonText.length];
MySpecialButton specialButton[]=new
MySpecialButton[specialButtonText.length];

Label displayLabel=new Label("0",Label.RIGHT);
Label memLabel=new Label(" ",Label.RIGHT);
```

```
final int FRAME_WIDTH=325,FRAME_HEIGHT=325;
final int HEIGHT=30, WIDTH=30,
H_SPACE=10,V_SPACE=10;
final int TOPX=30, TOPY=50;
MyCalculator(String frameText)//constructor
{
super(frameText);

int tempX=TOPX, y=TOPY;
displayLabel.setBounds(tempX,y,240,HEIGHT);
displayLabel.setBackground(Color.BLUE);
displayLabel.setForeground(Color.WHITE);
add(displayLabel);

memLabel.setBounds(TOPX,  TOPY+HEIGHT+
V_SPACE,WIDTH, HEIGHT);
add(memLabel);
  tempX=TOPX;
y=TOPY+2*(HEIGHT+V_SPACE);
for(int i=0; i<memoryButton.length; i++)
{
memoryButton[i]=new
MyMemoryButton(tempX,y,WIDTH,HEIGHT,memoryB
uttonText[i], this);
memoryButton[i].setForeground(Color.RED);
y+=HEIGHT+V_SPACE;
}
tempX=TOPX+1*(WIDTH+H_SPACE);
y=TOPY+1*(HEIGHT+V_SPACE);
for(int i=0;i<specialButton.length;i++)
{
```

```
specialButton[i]=new
MySpecialButton(tempX,y,WIDTH*2,HEIGHT,specialBu
ttonText[i], this);
specialButton[i].setForeground(Color.RED);
tempX=tempX+2*WIDTH+H_SPACE;
}
int digitX=TOPX+WIDTH+H_SPACE;
int digitY=TOPY+2*(HEIGHT+V_SPACE);
tempX=digitX;  y=digitY;
for(int i=0;i<digitButton.length;i++)
{
digitButton[i]=new
MyDigitButton(tempX,y,WIDTH,HEIGHT,digitButtonTe
xt[i], this);
digitButton[i].setForeground(Color.BLUE);
tempX+=WIDTH+H_SPACE;
if((i+1)%3==0){tempX=digitX; y+=HEIGHT+V_SPACE;}
}
int opsX=digitX+2*(WIDTH+H_SPACE)+H_SPACE;
int opsY=digitY;
tempX=opsX;  y=opsY;
for(int i=0;i<operatorButton.length;i++)
{
tempX+=WIDTH+H_SPACE;
operatorButton[i]=new
MyOperatorButton(tempX,y,WIDTH,HEIGHT,operatorB
uttonText[i], this);
operatorButton[i].setForeground(Color.RED);
if((i+1)%2==0){tempX=opsX; y+=HEIGHT+V_SPACE;}
}

addWindowListener(new WindowAdapter()
{
public void windowClosing(WindowEvent ev)
{System.exit(0);}
});

setLayout(null);
setSize(FRAME_WIDTH,FRAME_HEIGHT);
setVisible(true);
}
static String getFormattedText(double temp)
{
String resText=""+temp;
if(resText.lastIndexOf(".0")>0)
   resText=resText.substring(0,resText.length()-2);
return resText;
}
public static void main(String []args)
{
new MyCalculator("Calculator - JavaTpoint");
}
}
class MyDigitButton extends Button implements
ActionListener
{
MyCalculator cl;
MyDigitButton(int x,int y, int width,int height,String
cap, MyCalculator clc)
{
super(cap);
setBounds(x,y,width,height);
this.cl=clc;
this.cl.add(this);
```

```
addActionListener(this);
}
static boolean isInString(String s, char ch)
{
for(int i=0; i<s.length();i++) if(s.charAt(i)==ch) return
true;
return false;
}
public void actionPerformed(ActionEvent ev)
{
String
tempText=((MyDigitButton)ev.getSource()).getLabel();

if(tempText.equals("."))
{
 if(cl.setClear)
   {cl.displayLabel.setText("0.");cl.setClear=false;}
 else if(!isInString(cl.displayLabel.getText(),'.'))
   cl.displayLabel.setText(cl.displayLabel.getText()+".");
 return;
}

int index=0;
try{
      index=Integer.parseInt(tempText);
   }catch(NumberFormatException e){return;}

if (index==0 && cl.displayLabel.getText().equals("0"))
return;

if(cl.setClear)
```

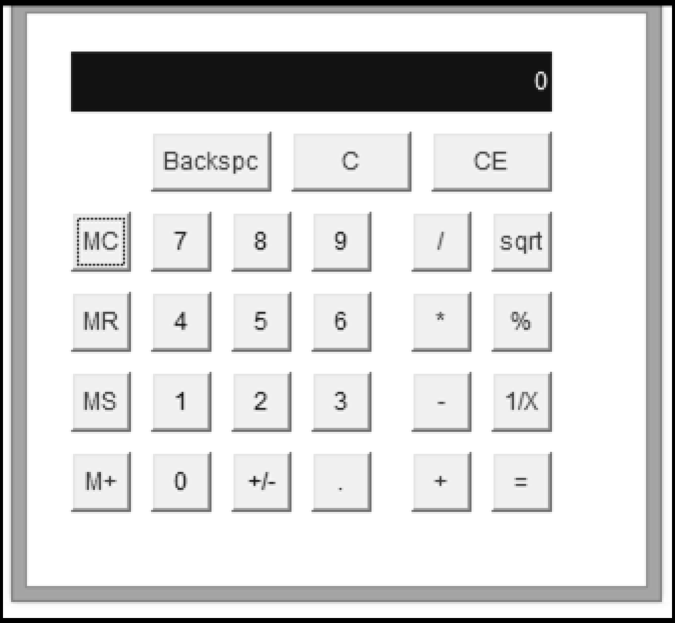
```
{cl.displayLabel.setText(""+index);cl.setClear=false;}
else

cl.displayLabel.setText(cl.displayLabel.getText()+index
);
}
}
class MyOperatorButton extends Button implements
ActionListener
{
MyCalculator cl;

MyOperatorButton(int x,int y, int width,int
height,String cap, MyCalculator clc)
{
super(cap);
setBounds(x,y,width,height);
this.cl=clc;
this.cl.add(this);
addActionListener(this);
}
public void actionPerformed(ActionEvent ev)
{
String
opText=((MyOperatorButton)ev.getSource()).getLabel(
);
cl.setClear=true;
double
temp=Double.parseDouble(cl.displayLabel.getText());
if(opText.equals("1/x"))
  {
   try
      {double tempd=1/(double)temp;

cl.displayLabel.setText(MyCalculator.getFormattedText
(tempd));}
   catch(ArithmeticException excp)
                  {cl.displayLabel.setText("Divide by 0.");}
   return;
   }
if(opText.equals("sqrt"))
   {
   try
      {double tempd=Math.sqrt(temp);

cl.displayLabel.setText(MyCalculator.getFormattedText
(tempd));}
         catch(ArithmeticException excp)
               {cl.displayLabel.setText("Divide by 0.");}
   return;
   }
if(!opText.equals("="))
   {
   cl.number=temp;
   cl.op=opText.charAt(0);
   return;
   }
switch(cl.op)
{
case '+':
   temp+=cl.number;break;
case '-':
   temp=cl.number-temp;break;
```

```
case '*':
   temp*=cl.number;break;
case '%':
   try{temp=cl.number%temp;}
   catch(ArithmeticException excp)
      {cl.displayLabel.setText("Divide by 0."); return;}
   break;
case '/':
   try{temp=cl.number/temp;}
      catch(ArithmeticException excp)
            {cl.displayLabel.setText("Divide by 0.");
return;}
   break;
}
cl.displayLabel.setText(MyCalculator.getFormattedText
(temp));  }
}
class MyMemoryButton extends Button implements
ActionListener
{
MyCalculator cl;
MyMemoryButton(int x,int y, int width,int height,String
cap, MyCalculator clc)
{
super(cap);
setBounds(x,y,width,height);
this.cl=clc;
this.cl.add(this);
addActionListener(this);
}
public void actionPerformed(ActionEvent ev)
{
```

```
char
memop=((MyMemoryButton)ev.getSource()).getLabel(
).charAt(1);

cl.setClear=true;
double
temp=Double.parseDouble(cl.displayLabel.getText());

switch(memop)
{
case 'C':
   cl.memLabel.setText(" ");cl.memValue=0.0;break;
case 'R':

cl.displayLabel.setText(MyCalculator.getFormattedText
(cl.memValue));break;
case 'S':
   cl.memValue=0.0;
case '+':

cl.memValue+=Double.parseDouble(cl.displayLabel.get
Text());
   if(cl.displayLabel.getText().equals("0") ||
cl.displayLabel.getText().equals("0.0")  )
      cl.memLabel.setText(" ");
   else
      cl.memLabel.setText("M");
   break;
}
}
}
```

```
class MySpecialButton extends Button implements
ActionListener
{
MyCalculator cl;

MySpecialButton(int x,int y, int width,int height,String
cap, MyCalculator clc)
{
super(cap);
setBounds(x,y,width,height);
this.cl=clc;
this.cl.add(this);
addActionListener(this);
}
static String backSpace(String s)
{
String Res="";
for(int i=0; i<s.length()-1; i++) Res+=s.charAt(i);
return Res;
}

public void actionPerformed(ActionEvent ev)
{
String
opText=((MySpecialButton)ev.getSource()).getLabel();
if(opText.equals("Backspc"))
{
String tempText=backSpace(cl.displayLabel.getText());
if(tempText.equals(""))
   cl.displayLabel.setText("0");
else
   cl.displayLabel.setText(tempText);
return;
}
if(opText.equals("C"))
{
cl.number=0.0; cl.op=' '; cl.memValue=0.0;
cl.memLabel.setText(" ");
}
  cl.displayLabel.setText("0");cl.setClear=true;
}
}
```

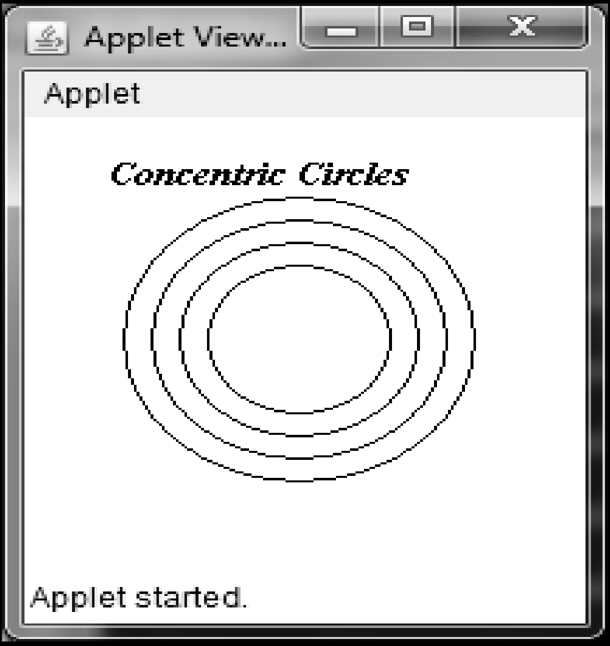
OUTPUT:



22. Write an applet to draw concentric circles in different colors.

-
  ```
importjava.awt.*;
importjava.applet.*;
public class ConcentricCircles extends Applet
{
public void paint( Graphics g)
{
g.setFont( new Font (“Times New Roman”, Font.BOLD|
Font.ITALIC,14));
g.drawString(“ Concentric Circles”, 30,30);
inti=65; j=65;
while (i>=30)
{
g.drawOval( i, i, j, j);
i = i – 10;
j = j + 20;
}
}
}
```

<u>OUTPUT</u>:



```
Applet started.
```

23. Design an applet to create a notepad.

➢
```
import java.io.*;
importjava.util.*;
importjava.swing.*;
importjava.awt.*;
importjava.awt.event.*;
importjava.applet.Applet.*;
public class notepad extends Keyadapter
implementsActionListener, KeyListener
{
staticint active = 0;
staticintfsize = 17;
JFrame frame1;
JMenunpMenuBar;
JMenu file, edit, format, view;
JMenuItemnewdoc, opendoc, exit, savedoc, saveasdoc,
copydoc, pastedoc, remdoc, fontfamily, fontstyle,
fontsize, status;
JTextAreamaintext;
JTextField title;
Font font1;
JPanel bottom;
JLabel details, pastecopydoc;
JListfamilylist, stylelist, sizelist;
JScrollPanesb;
String familyvalue[]={“Agency
FB”,”Antiqua”,Architect”,”Arial”,”Arial”,”Calibri”,”Comic
Sans”,”Courier”,”Cursive”,Impact”,”Serif”};
String
sizevalue[]={“5”,”10”,”15”,”20”,”25”,”30”,”35”,”40”,”45”,
”50”,”55”,”60”,”65”,”70”};
int [] stylevalue={ Font.PLAIN, Font.BOLD, Font.Italic};
String []  stylevalues={“PLAIN”,”BOLD”,”ITALIC”};
String ffamily, fsizestr, fstylestr;
intfstyle;
int cl;
```

```
intlinecount;
String tle;
String topicstitle =””;
JScrollPanesp;
notepad()
{
frame1 = new JFrame(“Notepad Fast”);
font1 = new font(“Arial”, Font.PLAIN,17);
bottom = new JPanel();
details = new JLabel();
pastecopydoc = new JLabel();
familylist = new JList(familyvalue);
stylelist = new JList(stylevalues);
sizelist = new JList(sizevalues);

familylist.setSelectionMode(ListSelectionModel.SINGLE
_SELECTION);
sizelist.setSelectionMode(ListSelectionModel.SINGLE_S
ELECTION);
stylelist.setSelectionMode(ListSelectionModel.SINGLE_
SELECTION);
bottom.add(details);
maintext = new JTextArea();
sp = new JScrollPane(maintext);
title = new JTextField(100);
sb = new JScrollPane(maintext);
maintext.setFont(font1);
frame1.add(maintext);
npMenuBar = new JMenuBar();
file = new JMenu(“FILE”);
edit = new JMenu(“EXIT”);
format = new JMenu(“FORMAT”);
view = new JMenu(“VIEW”);
newdoc = new JMenuItem(“New Document”);
opendoc = new JMenuItem(“Open Document”);
savedoc = new JMenuItem(“Save Document”);
saveasdoc = new JMenuItem(“Save As Document”);
```

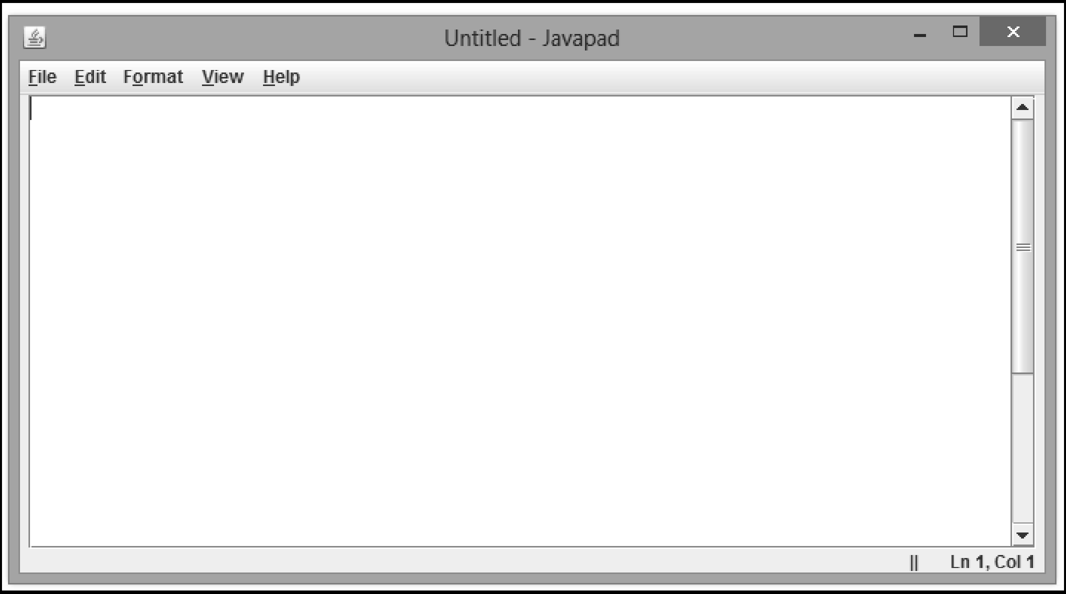
```
exit = new JMenuItem(“ Exit Document”);
copydoc = new JMenuItem(“Copy Document”);
remdoc = new JMenuItem(“Remove Document”);
pastedoc = new JMenuItem(“Paste Document”);
fontfamily = new JMenuItem(“Set Font Family”);
fontstyle = new JMenuItem(“Set Font Style”);
fontsize= new JMenuItem(“Set Font Size”);
status = new JMenuItem(“Status”);
file.add(newdoc);
file.add(opendoc);
file.add(savedoc);
file.add(saveasdoc);
file.add(exitdoc);
edit.add(copydoc);
edit.add(pastedoc);
edit.add(remdoc);
format.add(fontfamily);
format.add(fontstyle);
format.add(fontsize);
view.add(status);
npMenuBar.add(file);
npMenuBar.add(edit);
npMenuBar.add(format);
npMenuBar.add(view);
frame1.setJMenu(npMenuBar);
frame1.set(bottom, BorderLayout.SOUTH);
newdoc.addActionListener(this);
copydoc.addActionListener(this);
pastedoc.addActionListener(this);
remdoc.addActionListener(this);
status.addActionListener(this);
savedoc.addActionListener(this);
saveasdoc.addActionListener(this);
fontfamily.addActionListener(this);
fontsize.addActionListener(this);
fontstyle.addActionListener(this);
exit.addActionListener(this);
```

```
maintext.addKeyListener(this);
fame1.setSize(600, 600);
fame1.setDefaultCloseOperation(JFrame.EXIT_ON_CLO
SE);
frame1.setLocationRelative(null);
frame1.setVisible(true);
}
public void actionPerformed(ActionEvent ae)
{
if (ae.getSource()== newdoc)
{
frame1.setTitle("New Document.txt);
maintext.setText("");
}
else if ( ae.getSource()== copydoc)
{
String texts = maintext.getText();
pastecopydoc.setText(texts);
JOptionPane.showMessageDialog(null, "Copy
Text"+texts);
}
else if ( ae.getSource()== remdoc)
{
maintext.setText("");
JOptionPane.showMessageDialog(null, "Removed");
}
else if ( ae.getSource()== pastedoc)
{
if(maintext.getText().length()!=0)
{
maintext.setText(maintext.getText());
}
else
{
maintext.setText(pastecopydoc.getText());
}
}
```

```
else if ( ae.getSource()== status)
{
try
{
if(active==0)
{
File f= new File(tle+”.txt);
Details.setText(“Size : “+f.length());
}
}
catch(Exception e)
{
}
}
else if ( ae.getSource()== fontfamily)
{
JOptionPane.showConfirmDialog(null, familylist,
“Choose Font Family”,
JOptionPane.OK_CANCEL_OPTION,
JOptionPane.PLAIN_MESSAGE);
ffamily= String.valueOf(familylist.getSelectedValue());
font1=new Font(ffamily, fstyle, fsize);
maintext.setFont(font1);
}
else if ( ae.getSource()== fontstyle)
{
JOptionPane.showConfirmDialog(null, stylelist,
“Choose Font Style”, JOptionPane.OK_CANCEL_OPTION,
JOptionPane.PLAIN_MESSAGE);
fstyle = stylevalue[stylelist.getSelectedIndex()];
font1=new Font(ffamily, fstyle, fsize);
maintext.setFont(font1);
}
else if ( ae.getSource()== fontsize);
{
```

```
JOptionPane.showConfirmDialog(null, stylelist,
“Choose Font Size”, JOptionPane.OK_CANCEL_OPTION,
JOptionPane.PLAIN_MESSAGE);
fsizestr=String.valueOf(sizelist.getSelectedValue());
fsize = Interger.ParseInt(fsizestr);
font1= new Font(ffamily, fstyle, fsize);
maintext.setFont(font1);
}
else if ( ae.getSource()== exit);
{
frame1.dispose();
}
}
public void keyTyped(KeyEventke)
{
cl = maintext.getText().length();
linecount = maintext.getLineCount();
details.setText(“ Length: “+cl+” Line: “+linecount);
}
public static void main( String ar[])
{
new notepad();
}
}
```

OUTPUT:



REFERENCES

- ❖GEEKS FOR GEEKS
- ❖TUTORIAL POINT
- ❖JAVATPOINT.COM
- ❖YOUTUBE VIDEOS
- ❖PROGRAMIZ.COM
- ❖CLASS PRACTICAL
- ❖CLASS NOTES