JAVA PROGRAMMING

2021

NAME	:	AMAN KUMAR
ENROLLMENT No.	:	00176807720
BRANCH	:	INFORMATION TECHNOLOGY
YEAR/SEM	:	3RD Year/5TH Sem
SUBJECT	:	JAVA PROGRAMMING
SUBMITTED To.	:	MS.AMANDEEP KAUR

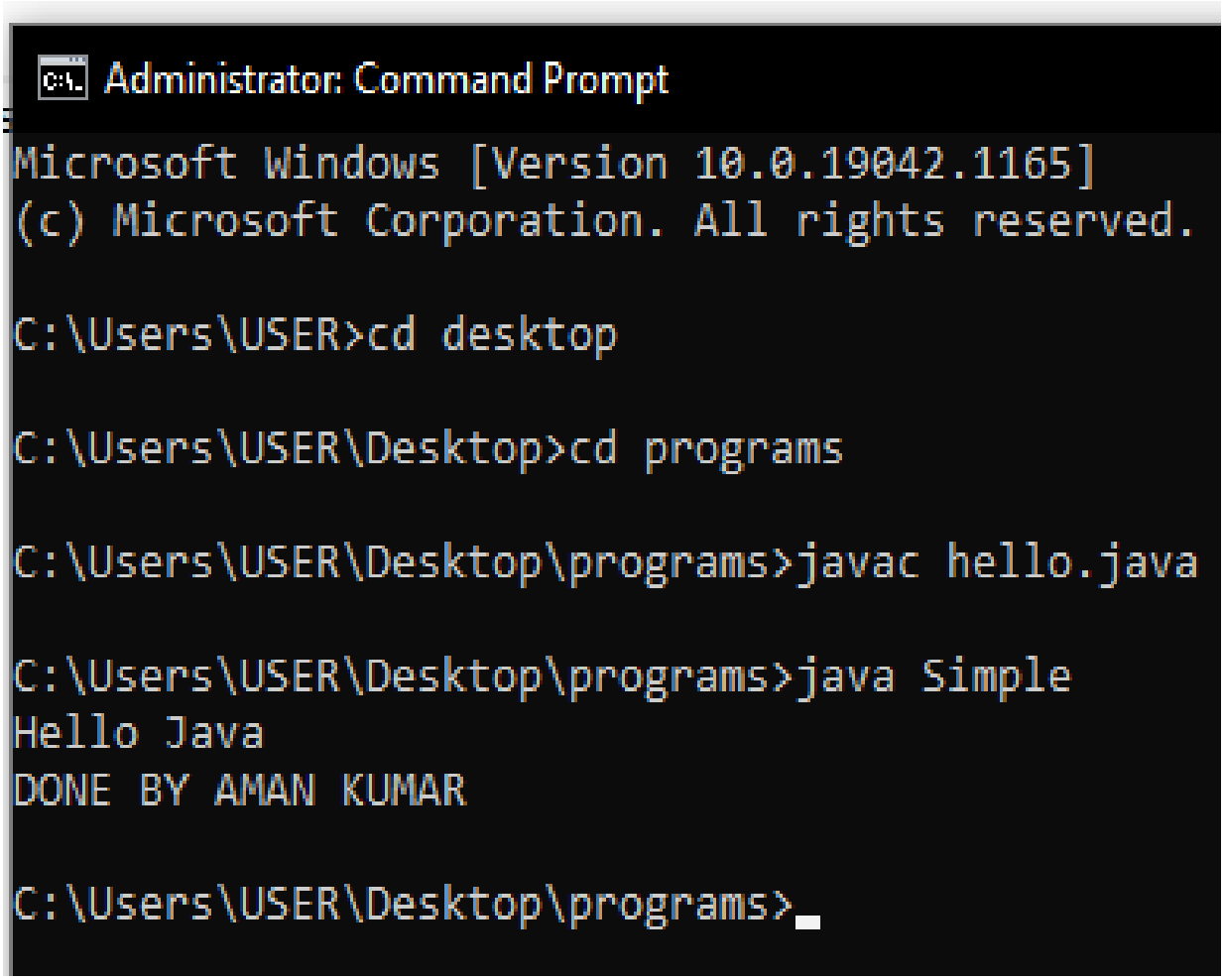
INDEX

PROGRAM NO:-01

Write a program to print hello world:-

```
class Simple{
    public static void main(String args[]){
     System.out.println ("Hello Java");
     System.out.println ("DONE BY AMAN KUMAR");

    }
}
```

OUTPUT:-



PROGRAM NO:-2

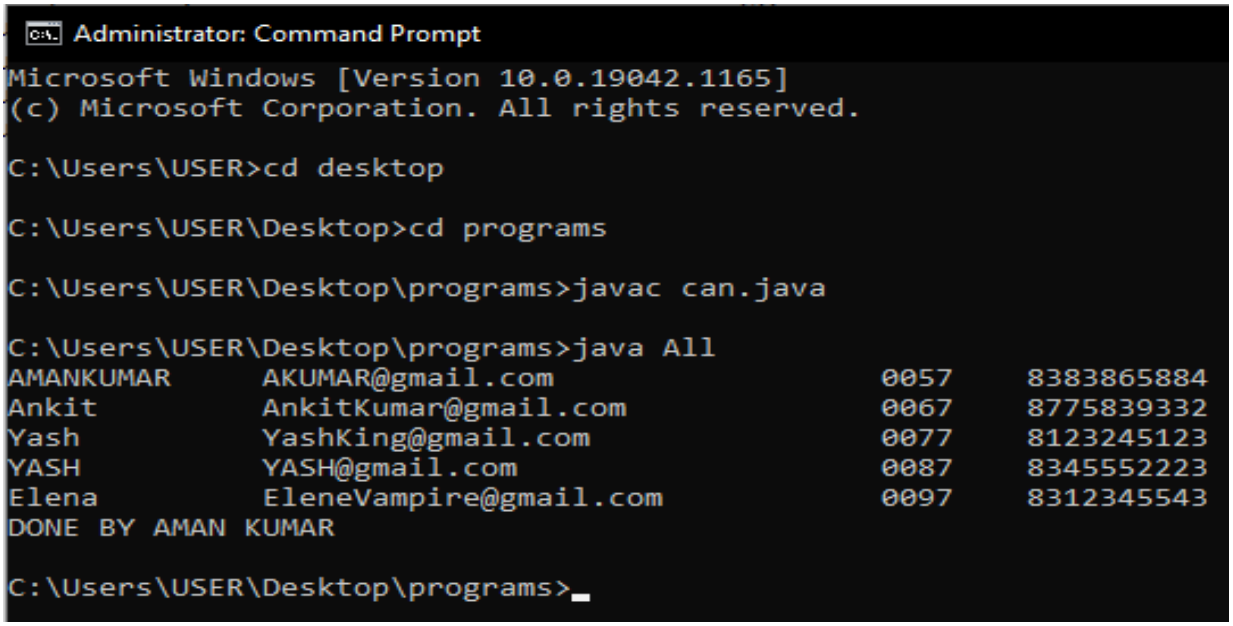
Write a program which prints the following about at least 5 person.

NAME EMAIL-ID EMPLOYEE_CODE PHONE

***each entry on a separate line.**

```
class All {

    public static void main(String[] args) {
        System.out.println("AMAN KUMAR " + "AKUMAR@gmail.com " + "0057 " +
"8383865884");
        System.out.println("Ankit " + "    AnkitKumar@gmail.com " + "     0067 " +
"8775839332");
        System.out.println("Yash " + "     YashKing@gmail.com " + "       0077 " +
"8123245123");
        System.out.println(" YASH " + "     YASH@gmail.com " + "   0087 " +
"8345552223");
        System.out.println("Elena " + "    EleneVampire@gmail.com " + "   0097 " +
"8312345543");
        System.out.println("DONE BY AMAN KUMAR");


    }
}
```

OUTPUT:

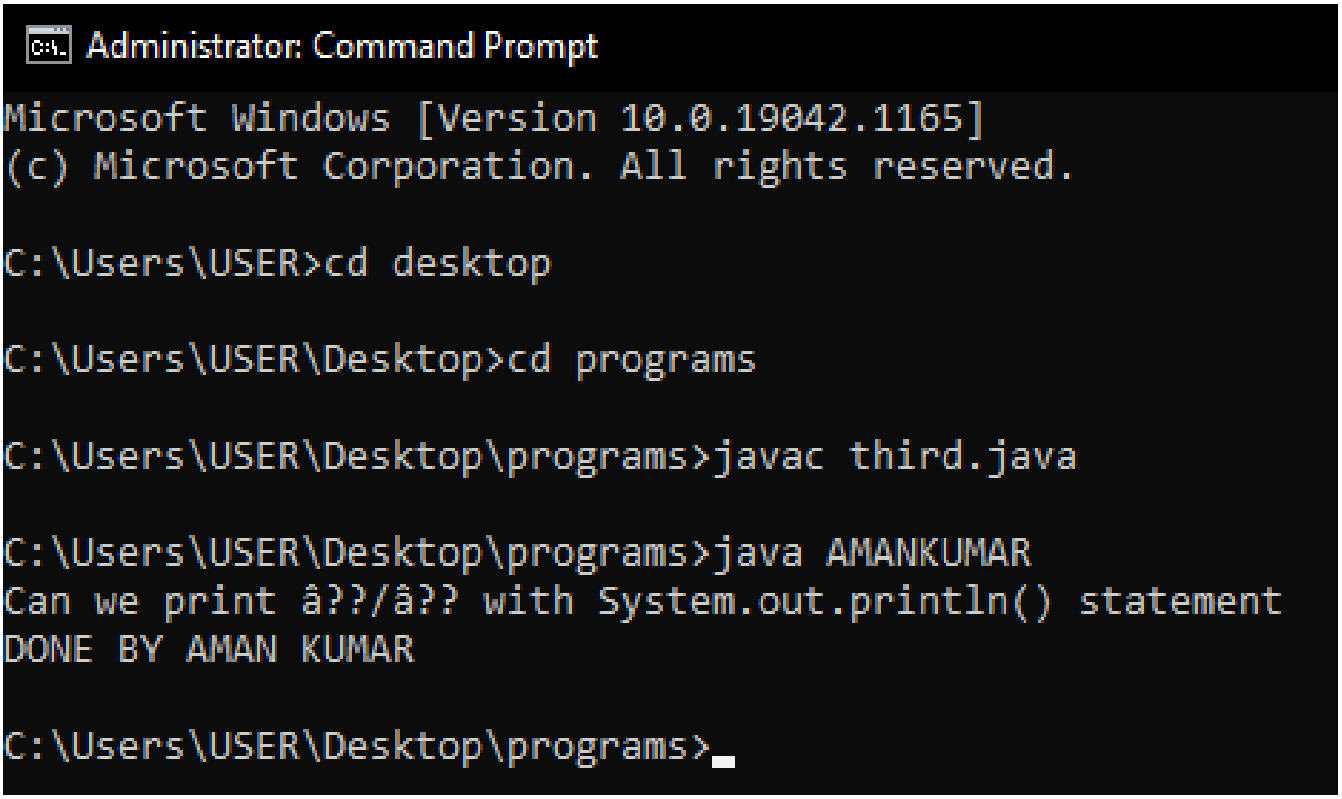


PROGRAM NO:-03

Write a program that prints the following line on the screen along the quotes
“Can we print ‘\’ with System.out.println() statement?”

```
class AMANKUMAR{
    public static void main(String args[]){
     System.out.println ("Can we print ‘\\’ with System.out.println() statement");
     System.out.println ("DONE BY AMAN KUMAR");

    }
}
```

OUTPUT:-

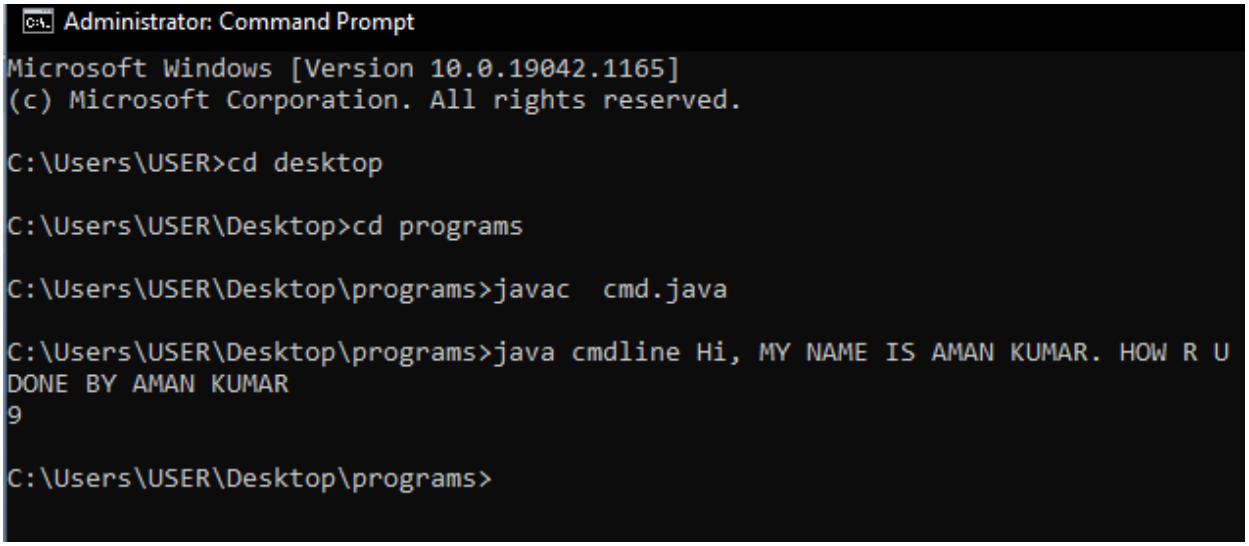


PROGRAM NO:-04

Write a program to count & display the no of command line arguments enter by the user.

```
class cmdline
 {
public static void main(String args[])
 {
System.out.println("DONE BY AMAN KUMAR");
int count;
count=args.length;
System.out.println(count);
 }
 }
```

Output:-

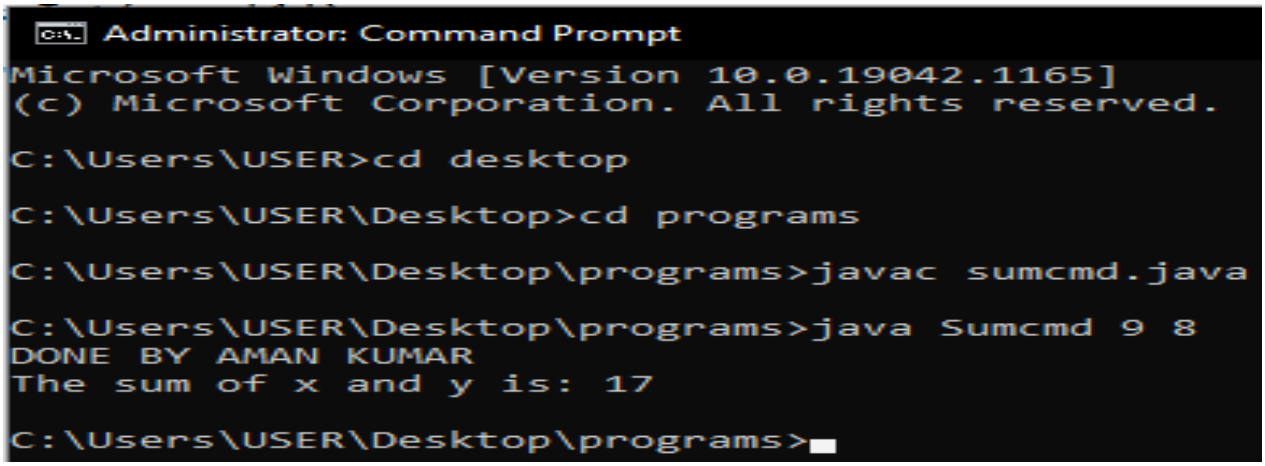


PROGRAM NO:-05

Write a program to find the sum of no's entered at command line arguments

```
class SumAtCmdLineArg
{
 public static void main(String args[]) {
 System.out.println("DONE BY AMAN KUMAR");
int x = Integer.parseInt(args[0]);
int y = Integer.parseInt(args[1]);
 int sum = x + y;
System.out.println("The sum of x and y is: " + sum);
 }
 }
```

OUTPUT:-

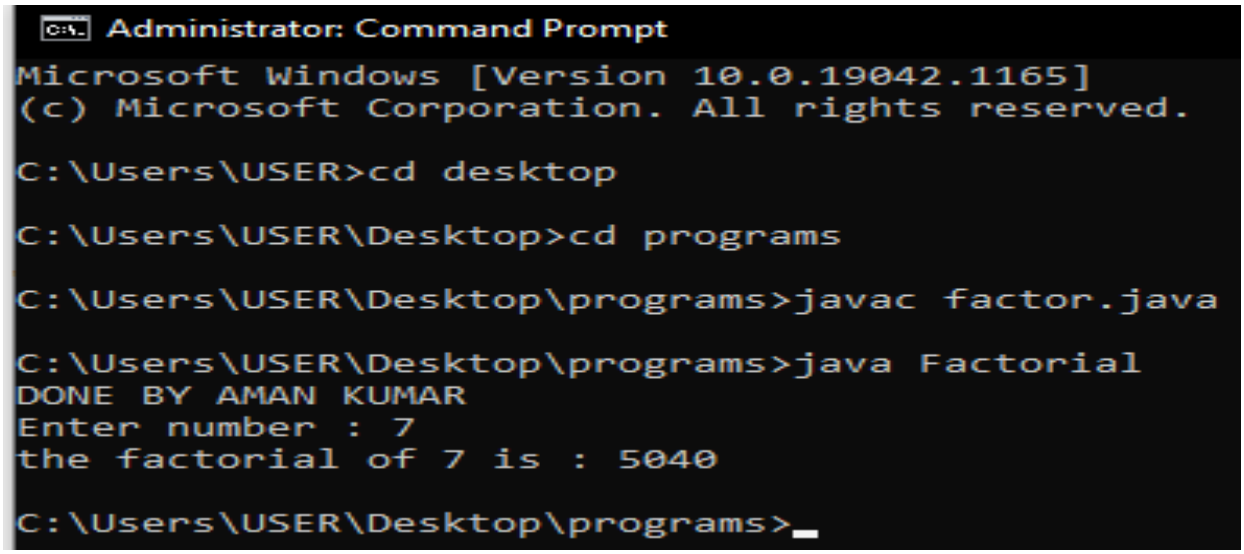


PROGRAM NO:-06

Write a program to calculate the factorial of a no enter by use.

```
import java.util.Scanner;
class Factorial
{
public static void main(String args[])
{
System.out.println("DONE BY AMAN KUMAR");
Scanner myObj = new Scanner(System.in);
System.out.print("Enter number : ");
int num = myObj.nextInt();
System.out.print("the factorial of " + num + " is : ");
long factorial = 1;
while(num > 0) {
factorial *= num;
num--;
}
System.out.println(factorial);
}
}
```

OUTPUT:-

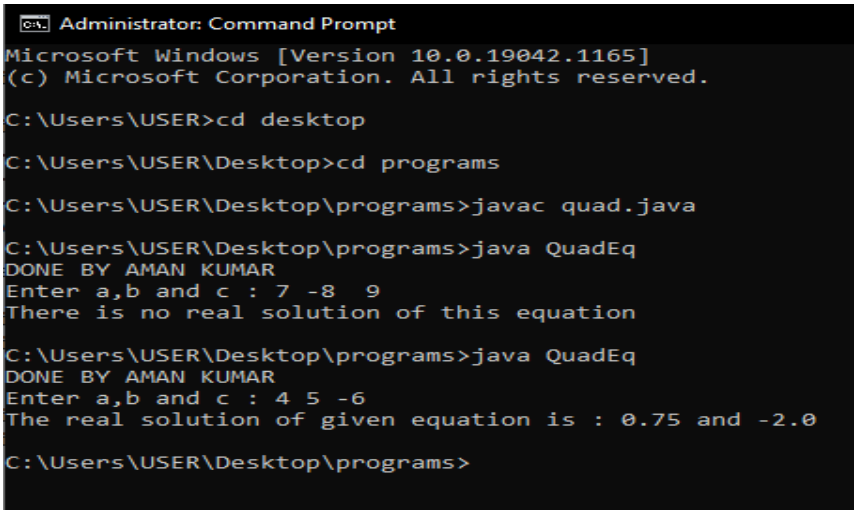


PROGRAM NO:-07

Write a program to print all the real soln to the quadratic eqn ax^2+bx+c=0. Display the message in case there are no real solns

```
import static java.lang.Math.*;
import java.util.Scanner;
class QuadEq
{
public static void main(String args[])
{
System.out.println("DONE BY AMAN KUMAR");
Scanner myObj = new Scanner(System.in);
System.out.print("Enter a,b and c : ");
int a = myObj.nextInt();
int b = myObj.nextInt();
int c = myObj.nextInt();
if (a == 0) System.out.println("Invalid");
int d = b * b - 4 * a * c;
double sqrt_val = sqrt(abs(d));
if (d > 0) {
System.out.print("The real solution of given equation is : ");
System.out.println( (-b + sqrt_val) / (2 * a) + " and " + (-b - sqrt_val) / (2 * a));
}
else if (d == 0) {
System.out.print("The real solution of given equation is : ");
System.out.println(-b / (2 * a) + " and " + -b / (2 * a));
}
else
{
System.out.println("There is no real solution of this equation");
}
}
}
```

OUTPUT:-

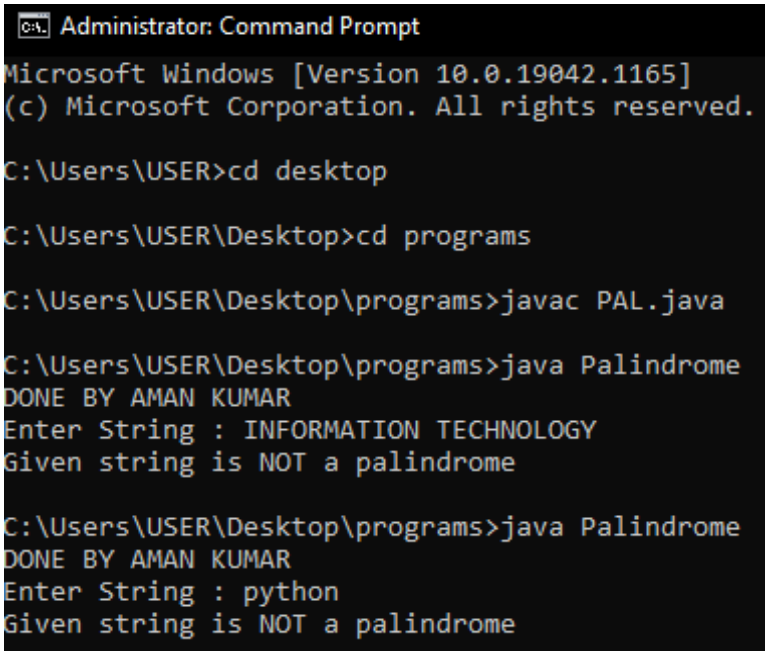


PROGRAM NO:-08

Write a program to check whether a string is palindrome or not.

```
import java.util.Scanner;
class Palindrome
{
public static void main(String args[])
{
System.out.println("DONE BY AMAN KUMAR");
Scanner myObj = new Scanner(System.in);
System.out.print("Enter String : ");
String str = myObj.nextLine();
int i = 0,j = str.length() - 1;
while(i < j) {
if(str.charAt(i) != str.charAt(j)) {
System.out.println("Given string is NOT a palindrome");
return ;
}
i++;
j--;
}
System.out.println("Given string is a palindrome");
}
}
```

OUTPUT:-

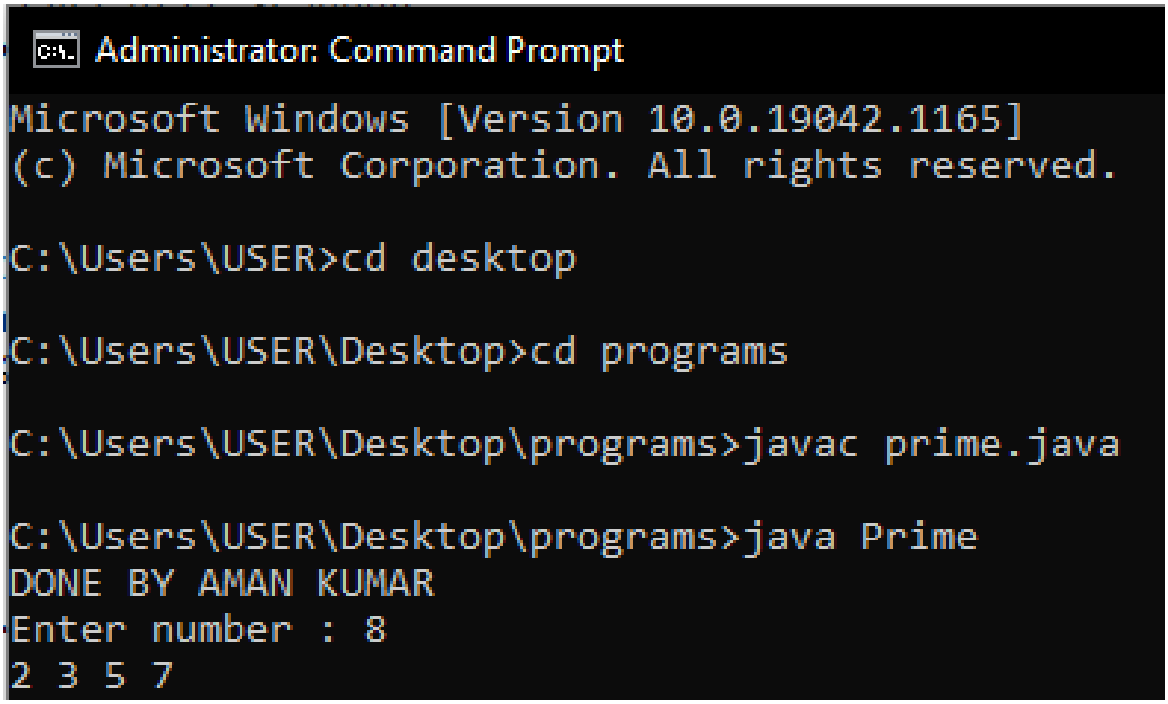


PROGRAM NO:-09

Write a program to print out all the prime no's upto the no entered by the user.

```
import java.util.Scanner;
class Prime
{
static boolean isPrime(int n)
{
if (n <= 1) return false;
for (int i = 2; i < n; i++)
if (n % i == 0) return false;
return true;
}
public static void main(String args[]) {
System.out.println("DONE BY AMAN KUMAR");
Scanner myObj = new Scanner(System.in);
System.out.print("Enter number : ");
int n = myObj.nextInt();
if(n == 1) {
System.out.print("There is no prime number upto " + n);
return ;
}
for (int i = 2; i <= n; i++)
if(isPrime(i) == true) System.out.print(i + " ");
}
}
```

OUTPUT:-

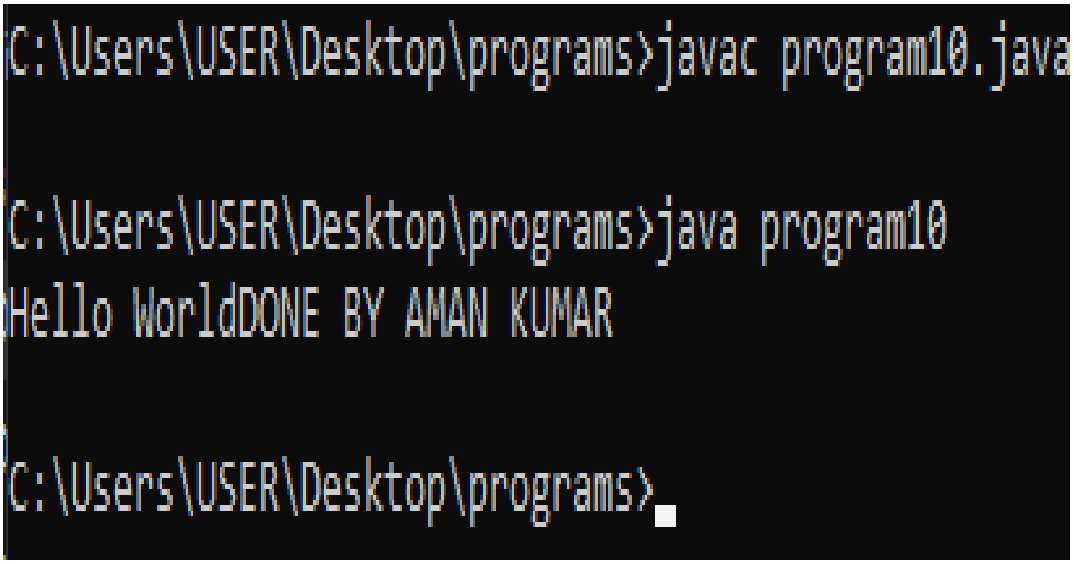


PROGRAM NO:-10

Write a program to print hello world without using semicolon.

```
class program10
{
   public static void main(String args[])
   {
      if (System.out.printf("Hello World") == null)
         {
         }
     System.out.println("DONE BY AMAN KUMAR");
   }
}
```

OUTPUT:-



PROGRAM NO:-11

Write a program to print Fibonacci series up to n term using Recursive and non Recursive

```
import java.util.*;
class program11
{
  static int n1=0,n2=1,n3=0;
    static void Fibonacci(int n)
    {
       if(n>=0){
           n3 = n1 + n2;
           n1 = n2;
           n2 = n3;
           System.out.print(" "+n3);
           Fibonacci(n-1);
       }
    }
       static void Fib(int n) {
       int firstTerm = 0;
       int secondTerm = 1;
       if (n <= 1) {
           System.out.println(n);
       } else {
           for (int i = 0; i <= n; ++i) {
               System.out.print(firstTerm + ", ");


               int nextTerm = firstTerm + secondTerm;
               firstTerm = secondTerm;
               secondTerm = nextTerm;
           }
       }

   }

   public static void main(String[] args) {
       Scanner sc = new Scanner(System.in);
       int n;
       int select;
       System.out.println("Enter the range of fibonacci:");
       n = sc.nextInt();
```

```
        System.out.println("-----MENU-----");
        System.out.println("1.Recursive");
        System.out.println("2.Non-Recursive");
        System.out.println("0.Exit");
        select = sc.nextInt();

        switch (select) {
            case 1:
            System.out.print(n1+" "+n2);
            Fibonacci(n-2);
                break;

            case 2:
                Fib(n);
                break;
            default:
                break;
        }
    System.out.println("FIBONACCI SERIES");
    System.out.println("DONE BY AMAN KUMAR");
    }
}
```

OUTPUT:-

```
C:\Users\USER\Desktop\programs>java program11
Enter the range of fibonacci:
8
-----MENU-----
1.Recursive
2.Non-Recursive
0.Exit
1
0 1 1 2 3 5 8 13 21FIBONACCI SERIES
DONE BY AMAN KUMAR

C:\Users\USER\Desktop\programs>java program11
Enter the range of fibonacci:
8
-----MENU-----
1.Recursive
2.Non-Recursive
0.Exit
2
0, 1, 1, 2, 3, 5, 8, 13, 21, FIBONACCI SERIES
DONE BY AMAN KUMAR
```

PROGRAM NO:-12

Write a program to define class students differentiate its objects in all possible ways. import java.io.*;

```
class cons1 {
    int num;
    String name;

    cons1() {
        System.out.println("Constructor called");
    }
}

class cons2 {
    cons2(String name) {
        System.out.println("Constructor with one " + "argument :
String : " + name);
    }


    cons2(String name, int age) {

        System.out.println("Constructor with two arguments : " + "
String and Integer : " + name + " " + age);

    }

    cons2(int id) {
        System.out.println("Constructor with one argument : " + " " +
id);
    }

}
```

```
class program12 {
    public static void main(String[] args) {

        cons1 a1 = new cons1();
        cons2 a2 = new cons2("AMAN KUMAR");
        cons2 a3 = new cons2("AMAN KUMAR", 21);
        cons2 a4 = new cons2(69);
        System.out.println("DONE BY AMAN KUMAR");
    }
}
```

OUTPUT:-

```
C:\Users\USER\Desktop\programs>javac program12.java

C:\Users\USER\Desktop\programs>java program12
Constructor called
Constructor with one argument : String : AMAN KUMAR
Constructor with two arguments :  String and Integer : AMAN KUMAR 21
Constructor with one argument :  69
DONE BY AMAN KUMAR
```

PROGRAM NO:-13

Write a program to implement stack.

```
import java.util.Stack;

class Program13
{
    public static void main(String[] args) {
        Stack<String> stack = new Stack<String>();

        stack.push("A");
        stack.push("B");
        stack.push("C");
        stack.push("D");

        System.out.println("The top element is " + stack.peek());

        stack.pop();
        stack.pop();

        System.out.println("The stack size is " + stack.size());

        if (stack.empty()) {
            System.out.println("The stack is empty");
        } else {
            System.out.println("The stack is not empty");
        }
        System.out.println("DONE BY AMAN KUMAR");
    }
}
```

OUTPUT

```
C:\Users\USER\Desktop\programs>javac program13.java

C:\Users\USER\Desktop\programs>java Program13
The top element is D
The stack size is 2
The stack is not empty
DONE BY AMAN KUMAR
```

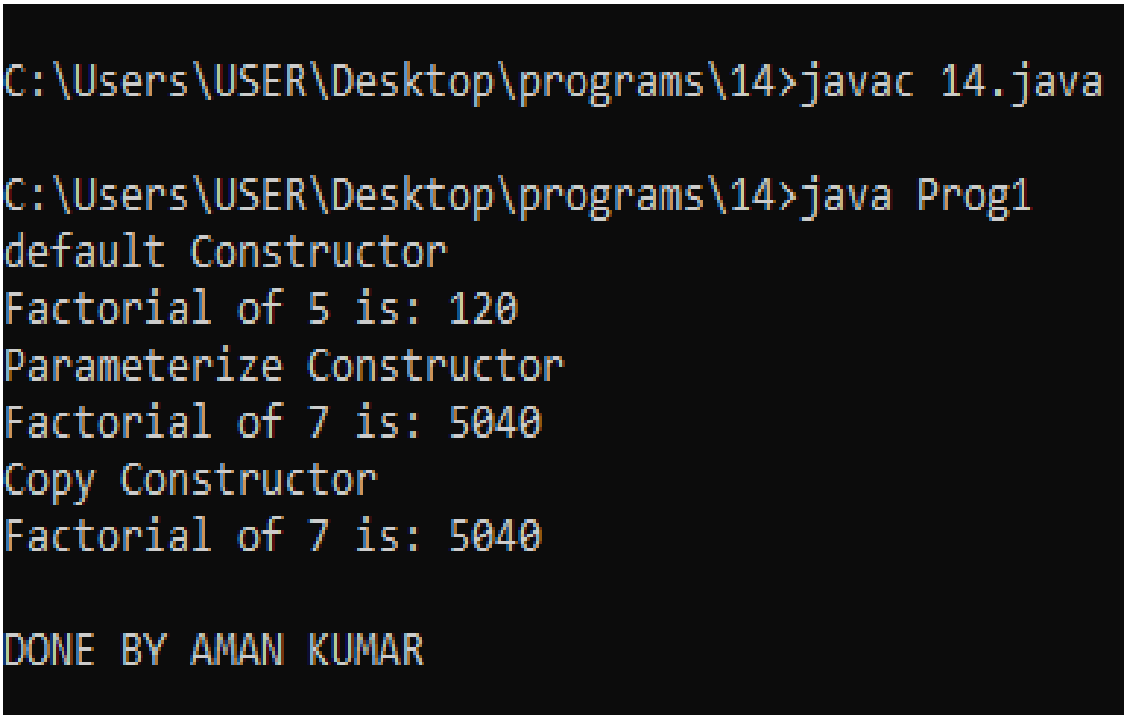
PROGRAM NO:-14

Write a program defining a class factorial. Define its constructor, overload the constructors and instantiate its object using all types of constructors.

```
import java.util.*;
class factorial {
 public int fact = 1;
 public int number;
 // Default Constructor
 public factorial() {
 System.out.println("default Constructor");
 this.number = 5;
 functionality();
 }
 // Parameterize Constructor
 public factorial(int num) {
 System.out.println("Parameterize Constructor");
 this.number = num;
 functionality(this.number);
 }
 // Copy Constructor
 public factorial(factorial obj) {
 System.out.println("Copy Constructor");
 this.number = obj.number;
 functionality(this.number);
 }
 // Function with no parameter
 public void functionality() {
 for (int i = 1; i <= 5; i++) {
 fact = fact * i;
 }
 System.out.println("Factorial of " + number + " is: " + fact);
 }
 // function with parameter
 public void functionality(int number) {
 for (int i = 1; i <= number; i++) {
 fact = fact * i;
 }
 System.out.println("Factorial of " + number + " is: " + fact);
 }
}
```

```
 class Prog1 {
 public static void main(String[] args) {
 factorial obj1 = new factorial();
 factorial obj2 = new factorial(7);
 factorial obj3 = new factorial(obj2);
 System.out.println(" ");
 System.out.println("DONE BY AMAN KUMAR");
 }
}
```

OUTPUT:-



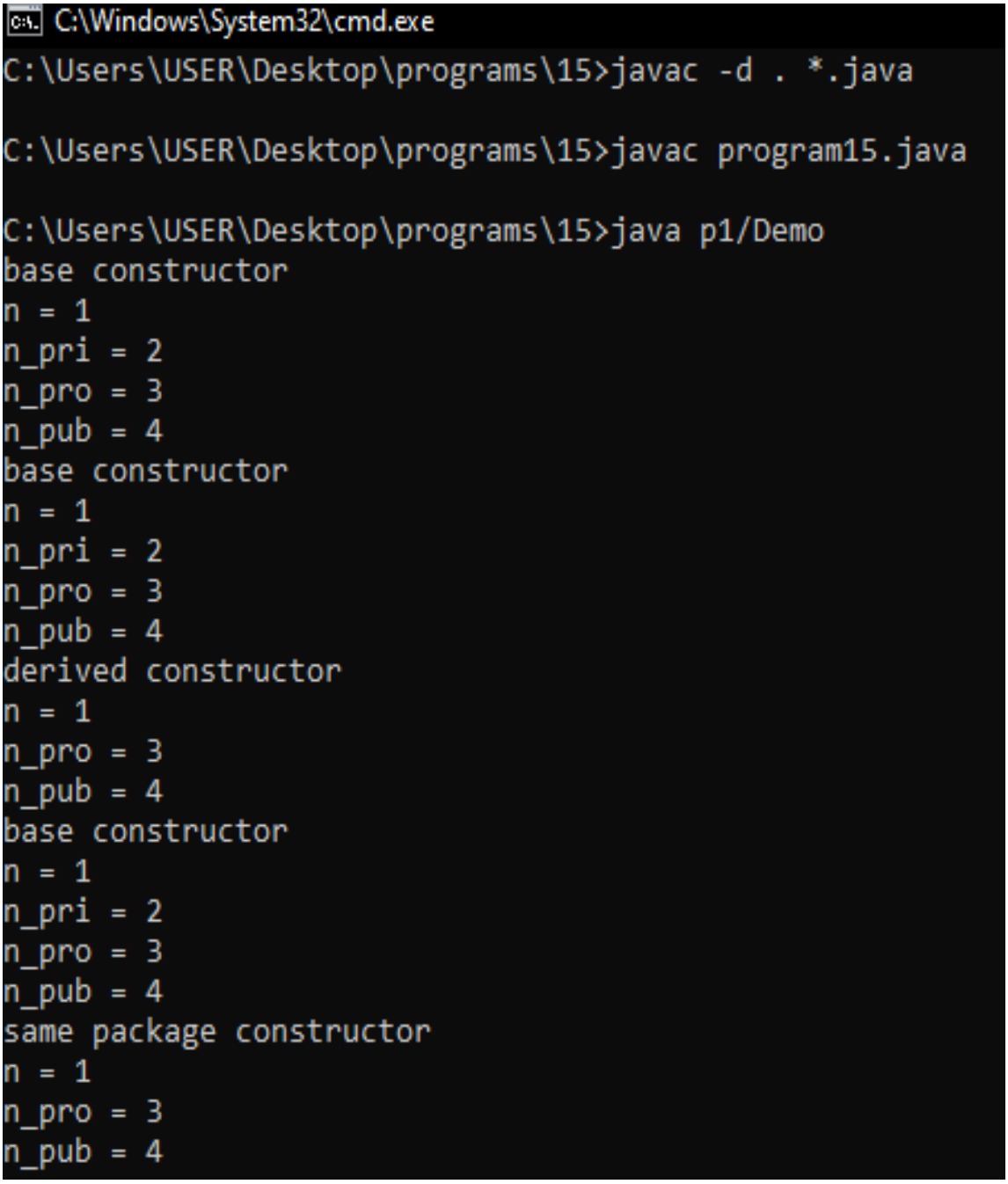
PROGRAM NO :-15

Write a program to Demonstrate access specifiers.

```
// Demo package p1.
package p1;
// Instantiate the various classes in p1.
 class Demo {
public static void main(String args[]) {
Protection ob1 = new Protection();
Derived ob2 = new Derived();
SamePackage ob3 = new SamePackage();
}
}

//package p1;
class Derived extends Protection {
Derived() {
System.out.println("derived constructor");
System.out.println("n = " + n);
// class only
// System.out.println("n_pri = "4 + n_pri);
System.out.println("n_pro = " + n_pro);
System.out.println("n_pub = " + n_pub);
}
}

//package p2;
class OtherPackage {
OtherPackage() {
p1.Protection p = new p1.Protection();
System.out.println("other package constructor");
// class or package only
// System.out.println("n = " + p.n);
// class only
// System.out.println("n_pri = " + p.n_pri);
// class, subclass or package only
// System.out.println("n_pro = " + p.n_pro);
System.out.println("n_pub = " + p.n_pub);
}
}

//package p1;
```

```
 class Protection {
int n = 1;
private int n_pri = 2;
protected int n_pro = 3;
public int n_pub = 4;
public Protection() {
System.out.println("base constructor");
System.out.println("n = " + n);
System.out.println("n_pri = " + n_pri);
System.out.println("n_pro = " + n_pro);
System.out.println("n_pub = " + n_pub);
}
}
//package p2;
class Protection2 extends p1.Protection {
Protection2() {
System.out.println("derived other package constructor");
// class or package only
// System.out.println("n = " + n);
// class only
// System.out.println("n_pri = " + n_pri);
System.out.println("n_pro = " + n_pro);
System.out.println("n_pub = " + n_pub);
}
}

//package p1;
class SamePackage {
SamePackage() {
Protection p = new Protection();
System.out.println("same package constructor");
System.out.println("n = " + p.n);
// class only
// System.out.println("n_pri = " + p.n_pri);
System.out.println("n_pro = " + p.n_pro);
System.out.println("n_pub = " + p.n_pub);
}
}
```

OUTPUT:-


C:\Windows\System32\cmd.exe
C:\Users\USER\Desktop\programs\15>javac -d . *.java
C:\Users\USER\Desktop\programs\15>javac program15.java
C:\Users\USER\Desktop\programs\15>java p1/Demo
base constructor
n = 1
n_pri = 2
n_pro = 3
n_pub = 4
base constructor
n = 1
n_pri = 2
n_pro = 3
n_pub = 4
derived constructor
n = 1
n_pro = 3
n_pub = 4
base constructor
n = 1
n_pri = 2
n_pro = 3
n_pub = 4
same package constructor
n = 1
n_pro = 3
n_pub = 4

PROGRAM NO:-16

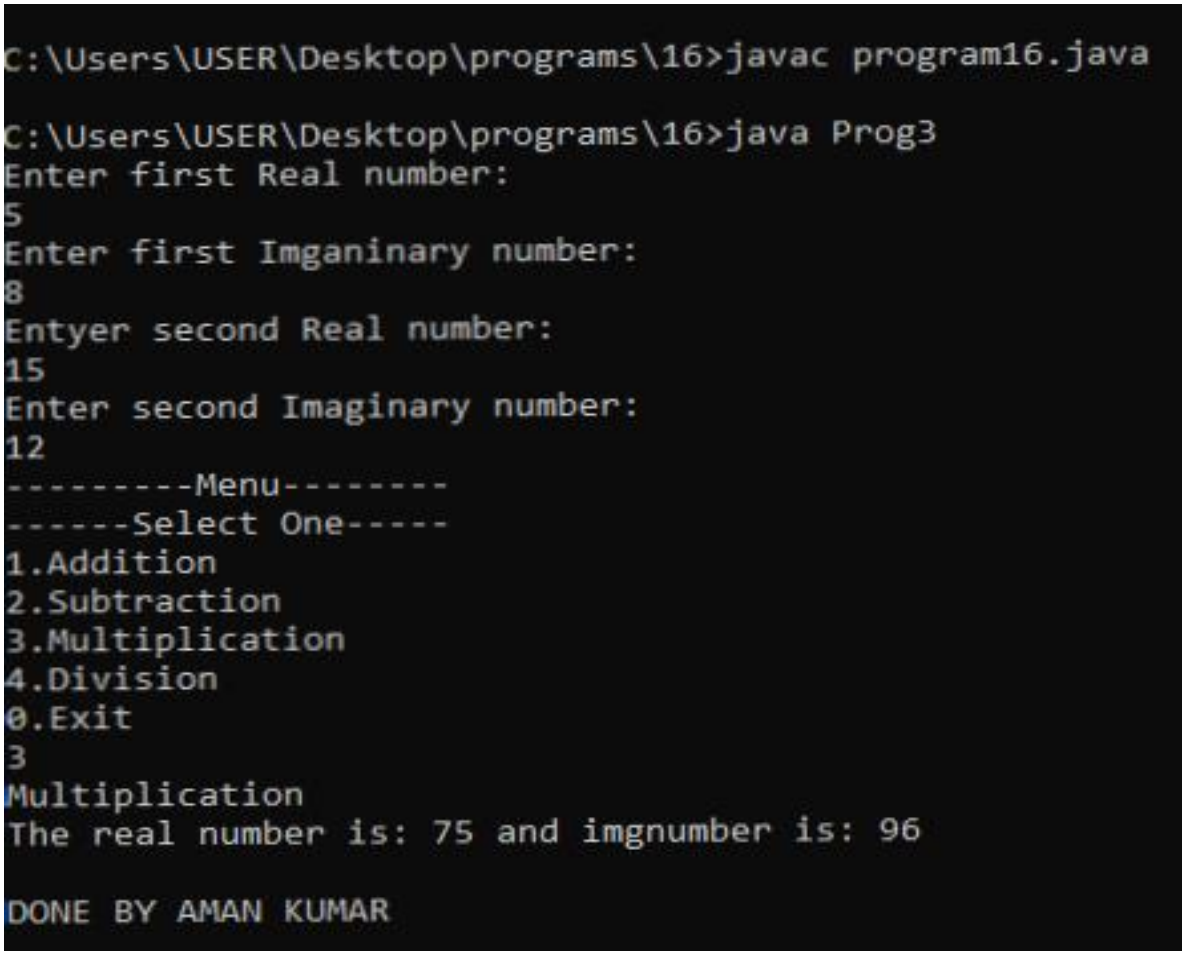
Write a complex class and perform the arithmetic operations on complex number. Use access a specifier
While creating a class. Compute by passing both objects to members methods of complex number.

```
import java.util.*;
class Complex {
 private int img, real;
 public Complex() {
 this.img = 10;
 this.real = 7;
 }
 public Complex(int real, int img) {
 this.real = real;
 this.img = img;
 }
 public void add(Complex obj1, Complex obj2) {
 this.real = obj1.real + obj2.real;
 this.img = obj1.img + obj2.img;
 }
 public void sub(Complex obj1, Complex obj2) {
 this.real = obj1.real - obj2.real;
 this.img = obj1.img - obj2.img;
 }
 public void mul(Complex obj1, Complex obj2) {
 this.real = obj1.real * obj2.real;
 this.img = obj1.img * obj2.img;
 }
 public void div(Complex obj1, Complex obj2) {
 this.real = obj1.real / obj2.real;
 this.img = obj1.img / obj2.img;
 }
 public void display() {
 System.out.println("The real number is: " + this.real + " and imgnumber is: " +
this.img);
 }
}
class Prog3 {
 public static void main(String[] args) {
 Scanner sc = new Scanner(System.in);
```

```
int select;
int real1, img1, real2, img2;
System.out.println("Enter first Real number: ");
real1 = sc.nextInt();
System.out.println("Enter first Imganinary number: ");
img1 = sc.nextInt();
System.out.println("Entyer second Real number: ");
real2 = sc.nextInt();
System.out.println("Enter second Imaginary number: ");
img2 = sc.nextInt();
Complex obj1 = new Complex(real1, img1);
Complex obj2 = new Complex(real2, img2);
Complex obj3 = new Complex();
System.out.println("---------Menu--------");
System.out.println("------Select One-----");
System.out.println("1.Addition");
System.out.println("2.Subtraction");
System.out.println("3.Multiplication");
System.out.println("4.Division");
System.out.println("0.Exit");
select = sc.nextInt();
switch (select) {
case 1:
System.out.println("Addition");
obj3.add(obj1, obj2);
obj3.display();
break;
case 2:
System.out.println("Subtraction");
obj3.sub(obj1, obj2);
obj3.display();
break;
case 3:
System.out.println("Multiplication");
obj3.mul(obj1, obj2);
obj3.display();
break;
case 4:
System.out.println("Division");
obj3.div(obj1, obj2);
obj3.display();
break;
default:
break;
```

```
 }
 System.out.println(" ");
 System.out.println("DONE BY AMAN KUMAR");
 }
}
```

OUTPUT



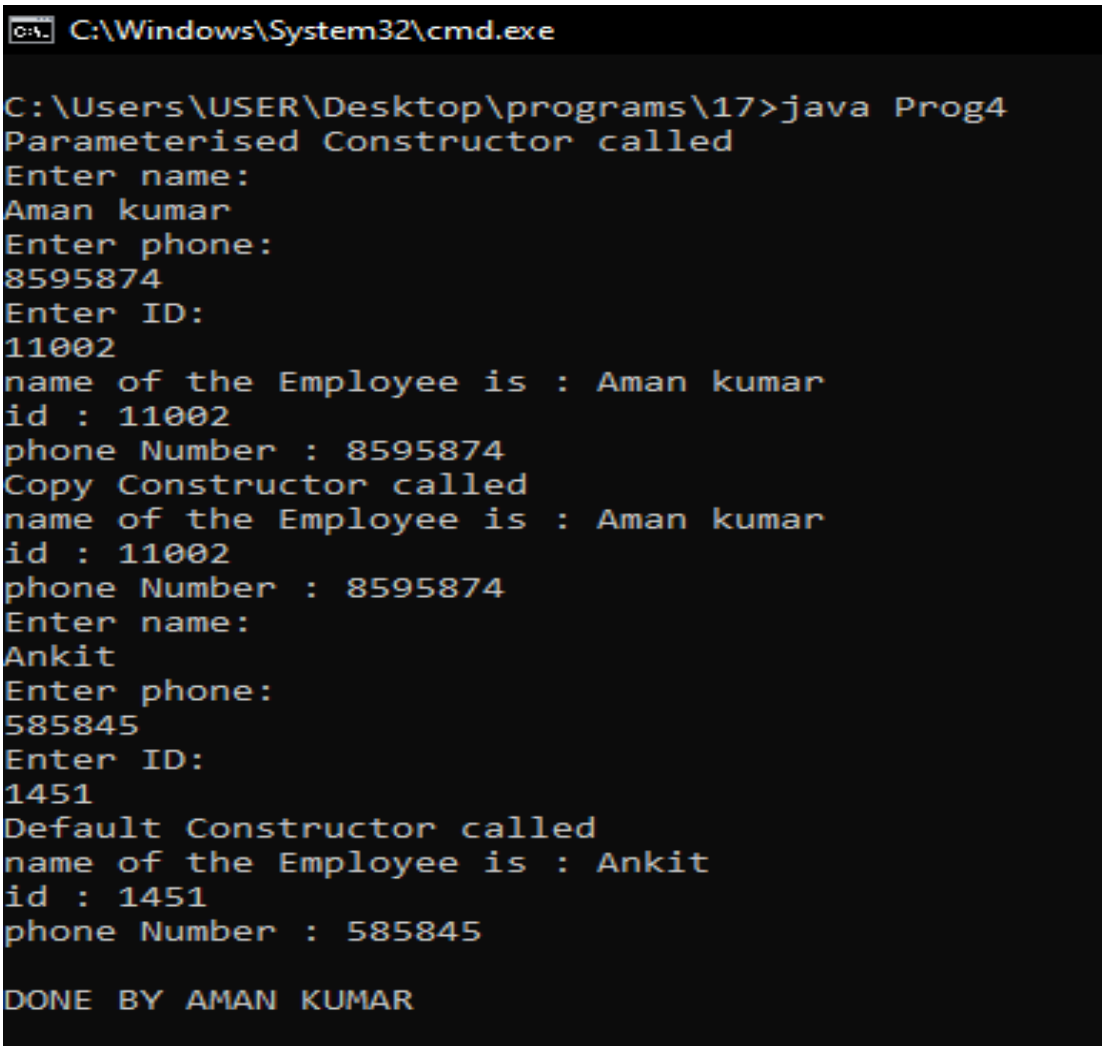
PROGRAM NO :-17

Write a program defining a class Employee and use all types of constructor to instantiate its objects. Also create a method that reads and displays the object content

```
import java.util.*;
class Employee {
 String name;
 long phone_num;
 int id;
 Employee() {
 System.out.println("Default Constructor called");
 name = null;
 phone_num = 0;
 id = 0;
 }
 Employee(String n, long p, int i) {
 name = n;
 phone_num = p;
 id = i;
 System.out.println("Parameterised Constructor called");
 }
 Employee(Employee emp) {
 this.name = emp.name;
 this.phone_num = emp.phone_num;
 this.id = emp.id;
 System.out.println("Copy Constructor called");
 }
 public void input(String n, long p, int i) {
 name = n;
 phone_num = p;
 id = i;
 }
 public void output() {
 System.out.println("name of the Employee is : " + name);
 System.out.println("id : " + id);
 System.out.println("phone Number : " + phone_num);
 }
}
class Prog4 {
 public static void main(String[] args) {
 String name;
 long Phone;
```

```
 int ID;
 Employee obj1 = new Employee("XYZ", 838386, 32);
 Scanner sc = new Scanner(System.in);
 System.out.println("Enter name: ");
 name = sc.nextLine();
 System.out.println("Enter phone: ");
 Phone = sc.nextInt();
 System.out.println("Enter ID: ");
 ID = sc.nextInt();
 obj1.input(name, Phone, ID);
 obj1.output();
 Employee obj2 = new Employee(obj1);
 obj2.output();
 Scanner sc2 = new Scanner(System.in);
 System.out.println("Enter name: ");
 name = sc2.nextLine();
 System.out.println("Enter phone: ");
 Phone = sc2.nextInt();
 System.out.println("Enter ID: ");
 ID = sc2.nextInt();
 Employee obj3 = new Employee();
 obj3.input(name, Phone, ID);
 obj3.output();
 System.out.println(" ");
 System.out.println("DONE BY AMAN KUMAR");
 }
}
```

OUTPUT:-


C:\Windows\System32\cmd.exe
C:\Users\USER\Desktop\programs\17>java Prog4
Parameterised Constructor called
Enter name:
Aman kumar
Enter phone:
8595874
Enter ID:
11002
name of the Employee is : Aman kumar
id : 11002
phone Number : 8595874
Copy Constructor called
name of the Employee is : Aman kumar
id : 11002
phone Number : 8595874
Enter name:
Ankit
Enter phone:
585845
Enter ID:
1451
Default Constructor called
name of the Employee is : Ankit
id : 1451
phone Number : 585845
DONE BY AMAN KUMAR

PROGRAM NO:-18

WAP to create a class Account that has the name of the customer, account number and type of account as data members. From this class derived the classes representing the 2 types of accounts 1. Current account 2. Saving account
The current account provides cheque facility with no interest.
The savings account provides compound interest but no cheque facility. All account holders should maintain a minimum balance and if the balance fall below this level a service charge is imposed. Include the necessary methods in order to achieve the following tasks.
a. Deposit the amount.
b. Withdraw the amount
c. Update the balance
d. Display the balance
e. Compute and deposit the interest
f. Check for the minimum balance and impose the penalty if necessary and update the balance.

```
import java.util.*;

class Account {
    private String accno;
    private String name;
    private String acc_type;
    private long balance;
    Scanner sc = new Scanner(System.in);

    // method to open new account
    public void openAccount() {
        System.out.print("Enter Account No: ");
        accno = sc.next();
        System.out.print("Enter Account type: ");
        acc_type = sc.next();
        System.out.print("Enter Name: ");
        name = sc.next();
        System.out.print("Enter Balance: ");
```

```
        balance = sc.nextLong();
    }

    // method to display account details
    public void showAccount() {
        System.out.println("\n");
        System.out.println("Name of account holder: " + name);
        System.out.println("Account no.: " + accno);
        System.out.println("Account type: " + acc_type);
        System.out.println("Balance: " + balance);
        System.out.println("\n");
    }

    // method to deposit money
    public void deposit() {
        long amt;
        System.out.println("Enter the amount you want to deposit:
");
        amt = sc.nextLong();
        balance = balance + amt;
    }

    public void deposit(long interest) {
        long amt;
        System.out.println("Enter the amount you want to deposit:
");
        amt = sc.nextLong();
        balance = balance + amt + interest;
    }

    // method to withdraw money
    public void withdrawal() {
        long amt;
        System.out.println("Enter the amount you want to withdraw:
");
```

```
        amt = sc.nextLong();
        if (balance >= amt) {
            balance = balance - amt;
            System.out.println("Balance after withdrawal: " +
balance);
        } else {
            System.out.println("Your balance is less than " + amt +
"\tTransaction failed...!!");
        }
    }

    // method to calculate the Interest
    public void Interest(int r, int t) {
        long A;
        int rate = 5;
        int T = 2;
        A = balance * (1 + rate * T);
        balance = balance + A;
        System.out.println("The Interest on your account is: " + A);

    }

    // method to search an account number
    public boolean search(String ac_no) {
        if (accno.equals(ac_no)) {
            showAccount();
            return (true);
        }
        return (false);
    }
}

class Prog18 {
    public static void main(String arg[]) {
        Scanner sc = new Scanner(System.in);
```

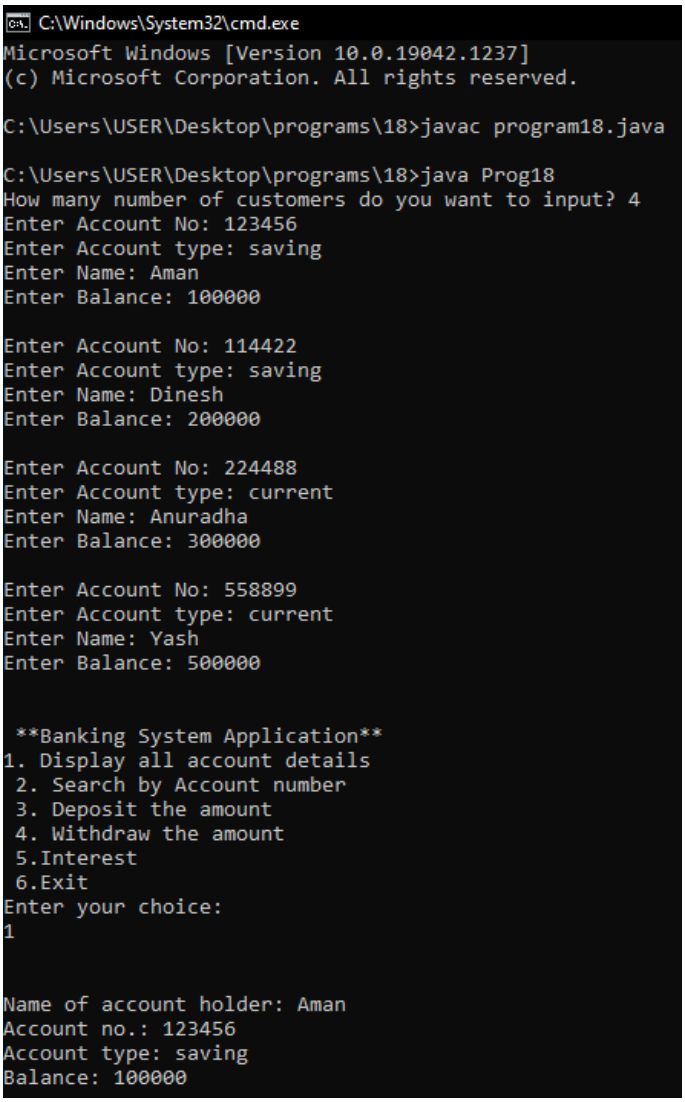
```
        // create initial accounts
        System.out.print("How many number of customers do you
want to input? ");
        int n = sc.nextInt();
        Account C[] = new Account[n];
        for (int i = 0; i < C.length; i++) {
            C[i] = new Account();
            C[i].openAccount();
            System.out.println(" ");
        }
        // loop runs until number 5 is not pressed to exit
        int ch;
        do {
            System.out.println("\n **Banking System Application**");
            System.out.println(
                    "1. Display all account details \n 2. Search by
Account number\n 3. Deposit the amount \n 4. Withdraw the
amount \n 5.Interest \n 6.Exit ");
            System.out.println("Enter your choice: ");
            ch = sc.nextInt();
            switch (ch) {
                case 1:
                    for (int i = 0; i < C.length; i++) {
                        C[i].showAccount();
                    }
                    break;
                case 2:
                    System.out.print("Enter account no. you want to
search: ");
                    String ac_no = sc.next();
                    boolean found = false;
                    for (int i = 0; i < C.length; i++) {
                        found = C[i].search(ac_no);
                        if (found) {
                            break;
```

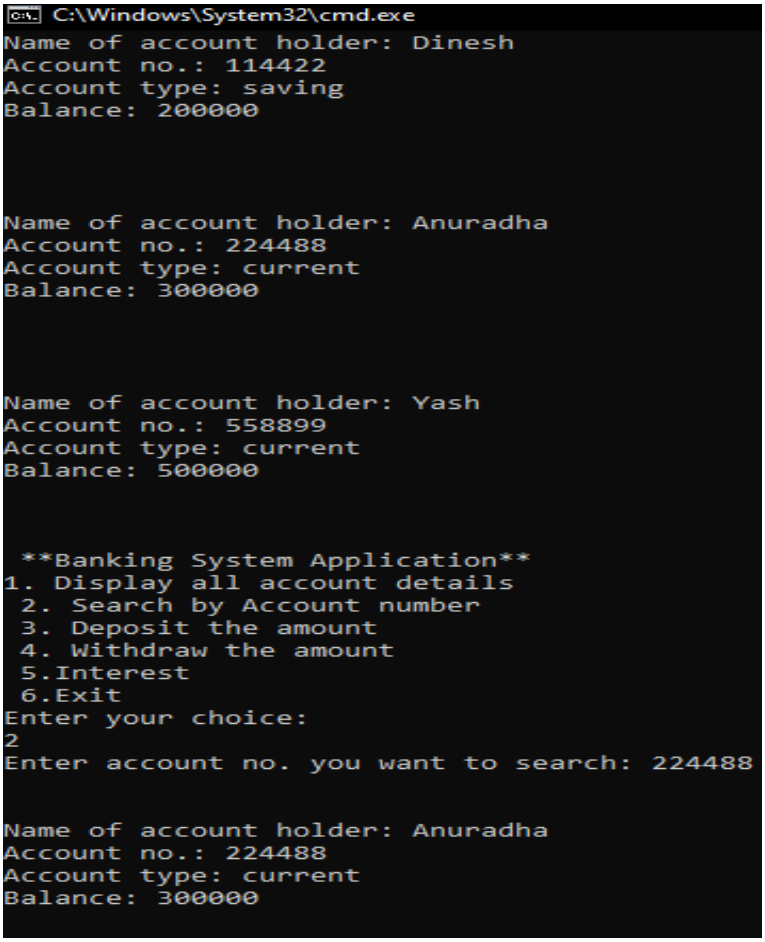
```
                }
            }
            if (!found) {
                System.out.println("Search failed! Account doesn't
exist..!!");
            }
            break;
        case 3:
            System.out.print("Enter Account no. : ");
            ac_no = sc.next();
            found = false;
            for (int i = 0; i < C.length; i++) {
                found = C[i].search(ac_no);
                if (found) {
                    C[i].deposit();
                    break;
                }
            }
            if (!found) {
                System.out.println("Search failed! Account doesn't
exist..!!");
            }
            break;
        case 4:
            System.out.print("Enter Account No : ");
            ac_no = sc.next();
            found = false;
            for (int i = 0; i < C.length; i++) {
                found = C[i].search(ac_no);
                if (found) {
                    C[i].withdrawal();
                    break;
                }
            }
            if (!found) {
```

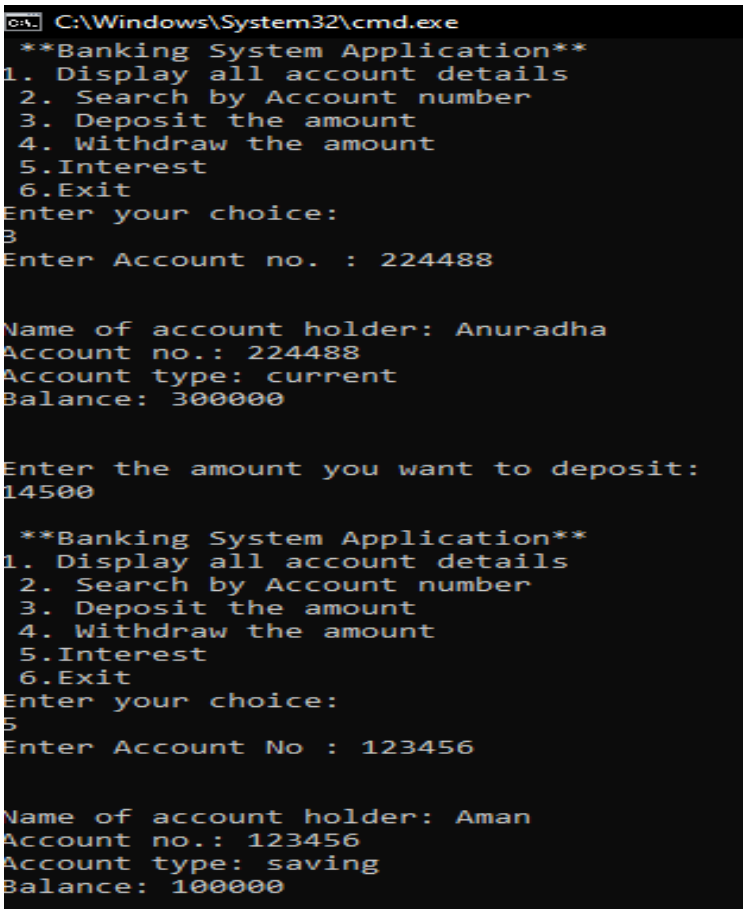
```
                System.out.println("Search failed! Account doesn't
exist..!!");
            }
            break;

        case 5:
            System.out.print("Enter Account No : ");
            ac_no = sc.next();
            found = false;
            for (int i = 0; i < C.length; i++) {
                found = C[i].search(ac_no);
                if (found) {
                    Scanner scs = new Scanner(System.in);
                    int rate, time;
                    System.out.println("Enter rate: ");
                    rate = scs.nextInt();
                    System.out.println("Enter Time: ");
                    time = scs.nextInt();
                    C[i].Interest(rate, time);
                    break;
                }
            }
            if (!found) {
                System.out.println("Search failed! Account doesn't
exist..!!");
            }
            break;
        case 6:
            System.out.println("See you soon...");
            break;
      }
    } while (ch != 6);
System.out.println(" ");
System.out.println("DONE BY AMAN KUMAR");
  }
```

}

OUTPUT:-


C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.
C:\Users\USER\Desktop\programs\18>javac program18.java
C:\Users\USER\Desktop\programs\18>java Prog18
How many number of customers do you want to input? 4
Enter Account No: 123456
Enter Account type: saving
Enter Name: Aman
Enter Balance: 100000
Enter Account No: 114422
Enter Account type: saving
Enter Name: Dinesh
Enter Balance: 200000
Enter Account No: 224488
Enter Account type: current
Enter Name: Anuradha
Enter Balance: 300000
Enter Account No: 558899
Enter Account type: current
Enter Name: Yash
Enter Balance: 500000
Banking System Application
1. Display all account details
2. Search by Account number
3. Deposit the amount
4. Withdraw the amount
5.Interest
6.Exit
Enter your choice:
1
Name of account holder: Aman
Account no.: 123456
Account type: saving
Balance: 100000


C:\Windows\System32\cmd.exe
Name of account holder: Dinesh
Account no.: 114422
Account type: saving
Balance: 200000
Name of account holder: Anuradha
Account no.: 224488
Account type: current
Balance: 300000
Name of account holder: Yash
Account no.: 558899
Account type: current
Balance: 500000
Banking System Application
1. Display all account details
2. Search by Account number
3. Deposit the amount
4. Withdraw the amount
5.Interest
6.Exit
Enter your choice:
2
Enter account no. you want to search: 224488
Name of account holder: Anuradha
Account no.: 224488
Account type: current
Balance: 300000


C:\Windows\System32\cmd.exe
Banking System Application
1. Display all account details
2. Search by Account number
3. Deposit the amount
4. Withdraw the amount
5.Interest
6.Exit
Enter your choice:
3
Enter Account no. : 224488
Name of account holder: Anuradha
Account no.: 224488
Account type: current
Balance: 300000
Enter the amount you want to deposit:
14500
Banking System Application
1. Display all account details
2. Search by Account number
3. Deposit the amount
4. Withdraw the amount
5.Interest
6.Exit
Enter your choice:
5
Enter Account No : 123456
Name of account holder: Aman
Account no.: 123456
Account type: saving
Balance: 100000

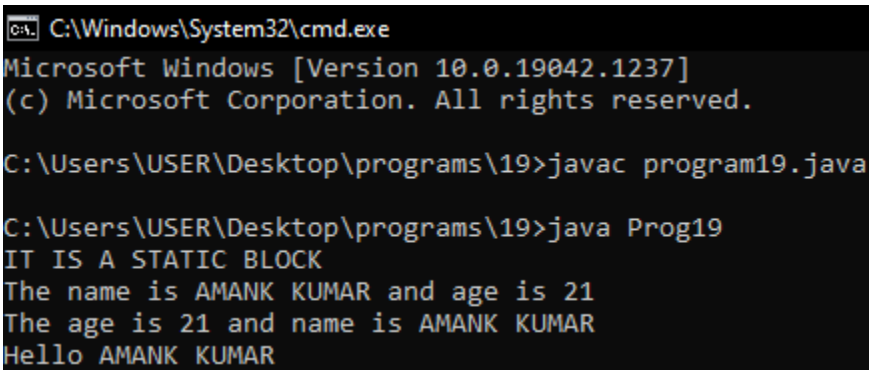
```
Enter rate:
5
Enter Time:
5
The Interest on your account is: 1100000

 **Banking System Application**
1. Display all account details
 2. Search by Account number
 3. Deposit the amount
 4. Withdraw the amount
 5.Interest
 6.Exit
Enter your choice:
6
See you soon...

DONE BY AMAN KUMAR
```

PROGRAM NO:-19

WAP to implement static block, static variable, static method and static nested class.

```
class Prog19 {
    // static variable
    static int Age = 21;
    static String name = "AMANK KUMAR";
    public static void Show() {
        System.out.println("The age is " + Age + " and name is " + name);
    }
    static {
        System.out.println("IT IS A STATIC BLOCK ");
    }
    static class Inside {
        public void Greet() {
            System.out.println("Hello " + name);
        }
    }

    public static void main(String[] args) {
        System.out.println("The name is " + name + " and age is " + Age);
        Show();

        Prog19.Inside obj = new Prog19.Inside();
        obj.Greet();
    }
}
```

OUTPUT:-



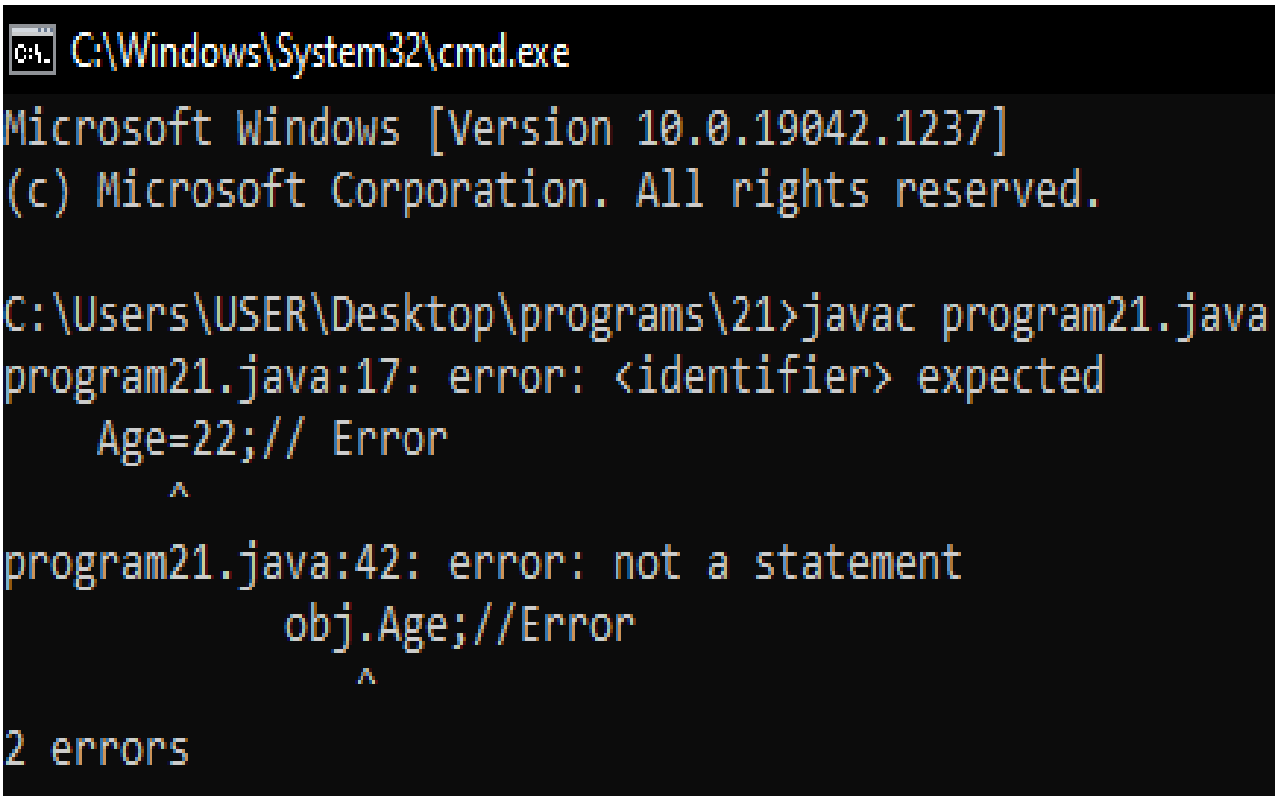
PROGRAM NO:-20

WAP to demonstrate final keyword.

```
class base {
    final int Age = 21;
    public void show() {
        System.out.println("Age is " + Age);
    }
    final public void Display() {
        System.out.println("Final Method");
        System.out.println("Your age is " + Age);
    }
}
final class child extends base {
    public String name = "AMAN KUMAR";
    Age=22;// Error
    public void show() {
        System.out.println("Name is " + name);
    }
    // It gives error
    void Display() {
        System.out.println("Final Method in derived class");
    }
}
// Error here cannot inherit from final class
class child2 extends child {
}
public class Prog21 {
    public static void main(String[] args) {
        base obj = new base();
```

```
        obj.show();

        child obj2 = new child();
            obj2.Display();//Error
            obj.Age;//Error

System.out.println(" ");

System.out.println("DONE BY AMAN KUMAR");
    }
}
```

OUTPUT:-



PROGRAM NO:-21

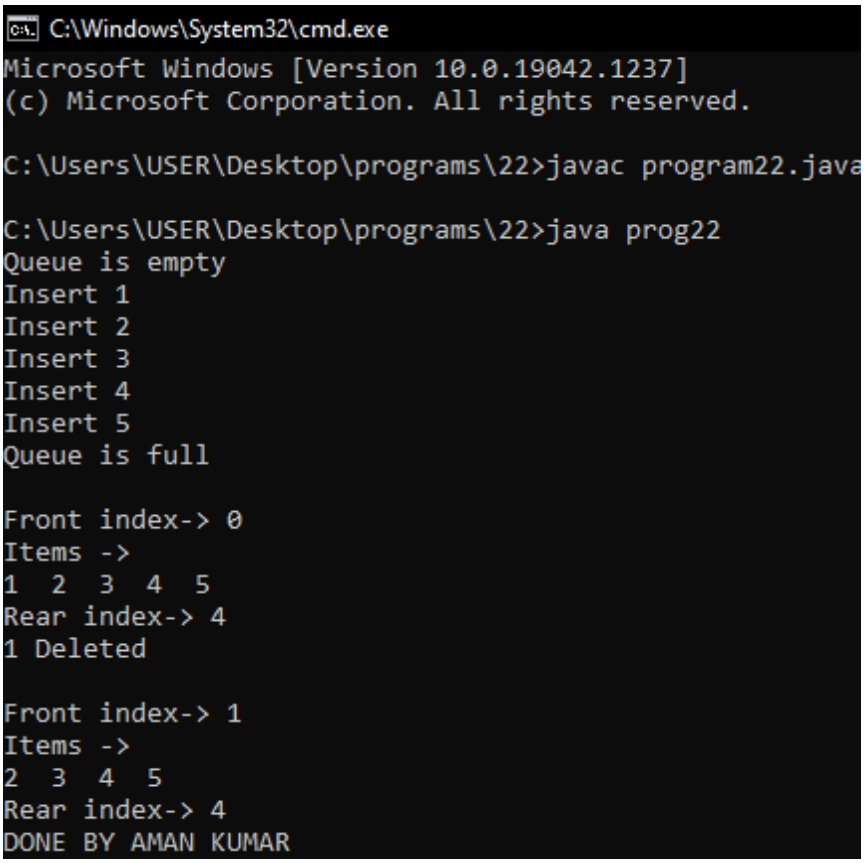
WAP to Implement Queue.

```
class prog22 {
  int SIZE = 5;
  int items[] = new int[SIZE];
  int front, rear;

  prog22() {
    front = -1;
    rear = -1;
  }

  // check if the queue is full
  boolean isFull() {
    if (front == 0 && rear == SIZE - 1) {
      return true;
    }
    return false;
  }

  // check if the queue is empty
  boolean isEmpty() {
    if (front == -1)
      return true;
    else
      return false;
  }

  // insert elements to the queue
  void enQueue(int element) {

    // if queue is full
    if (isFull()) {
      System.out.println("Queue is full");
    }
    else {
      if (front == -1) {
        // mark front denote first element of queue
        front = 0;
      }
```

```
    rear++;
    // insert element at the rear
    items[rear] = element;
    System.out.println("Insert " + element);
  }
}
// delete element from the queue
int deQueue() {
  int element;
  // if queue is empty
  if (isEmpty()) {
    System.out.println("Queue is empty");
    return (-1);
  }
  else {
    element = items[front];
    if (front >= rear) {
      front = -1;
      rear = -1;
    }
    else {
      // mark next element as the front
      front++;
    }
    System.out.println( element + " Deleted");
    return (element);
  }
}
// display element of the queue
void display() {
  int i;
  if (isEmpty()) {
    System.out.println("Empty Queue");
  }
  else {

    System.out.println("\nFront index-> " + front);

    System.out.println("Items -> ");
    for (i = front; i <= rear; i++)
      System.out.print(items[i] + "  ");
    System.out.println("\nRear index-> " + rear);
  }
```

```
  }
  public static void main(String[] args) {
    prog22 q = new prog22();
    q.deQueue();
    // insert elements to the queue
    for(int i = 1; i < 6; i ++) {
      q.enQueue(i);
    }
    // 6th element can't be added to queue because queue is full
    q.enQueue(6);
    q.display();
    // deQueue removes element entered first i.e. 1
    q.deQueue();
    // Now we have just 4 elements
    q.display();
System.out.println("DONE BY AMAN KUMAR");

  }
}
```

OUTPUT:-



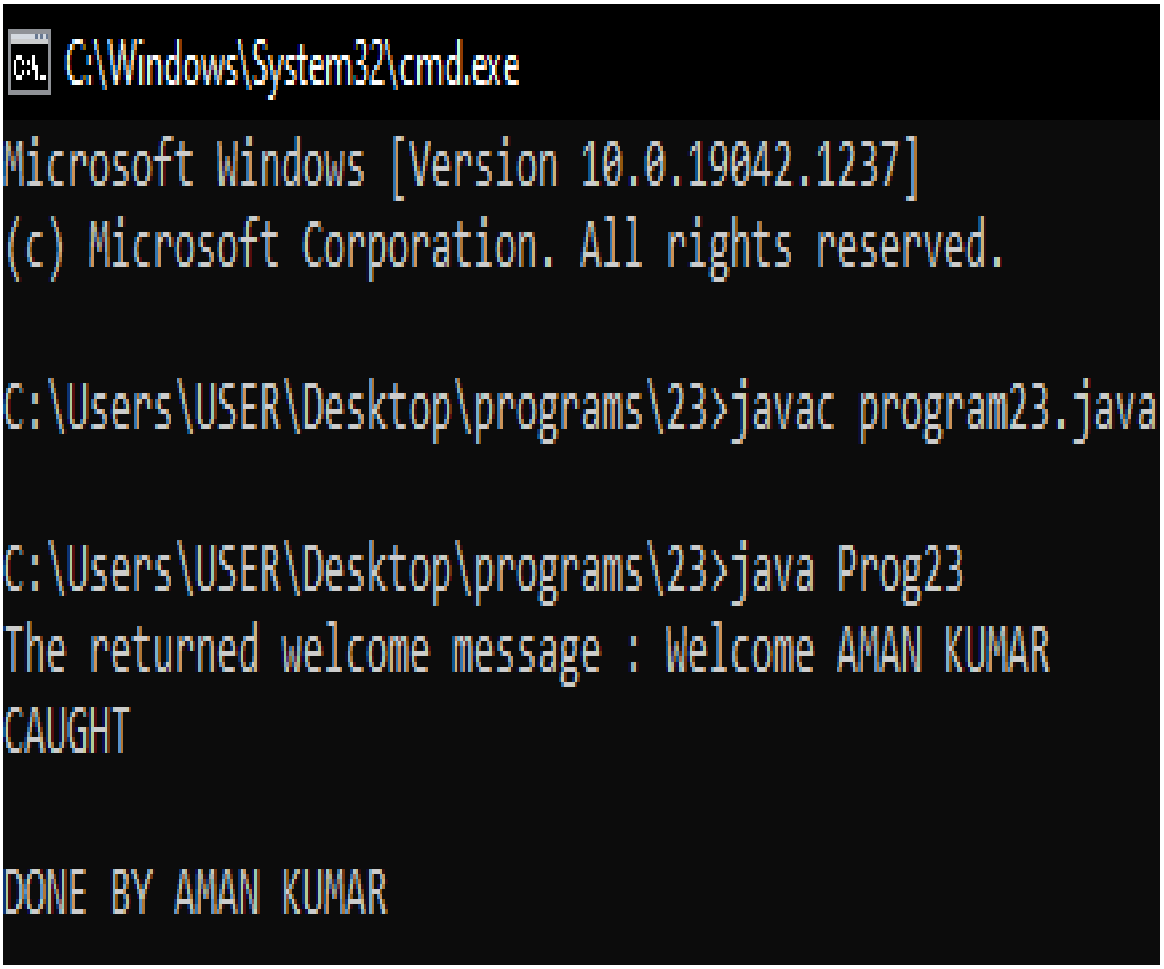
PROGRAM NO:-22

To create a customized exception and also make use of all the five exception keywords.

```
class MyException extends Exception
{
   public MyException(String s) {
      super(s);
   }
}

public class Prog23{

   public static void main(String[] args) {

      try {

         String welcomeMessage = welcomeMessage("AMAN KUMAR");

         System.out.println("The returned welcome message : " +
welcomeMessage);
      }
      catch (NullPointerException npex){
         System.out.println("Exception handled : " + npex.toString());
      }
      finally {
         System.out.println("CAUGHT");


System.out.println(" ");
System.out.println("DONE BY AMAN KUMAR");
      }
   }

   public static String welcomeMessage(String name)
         throws NullPointerException {

      if(name == null) {

         throw new NullPointerException("Invoke method with VALID name");
      }
```

```
        String welcomeMsg = "Welcome " + name;

        return welcomeMsg;

    }
}
```

OUTPUT:-

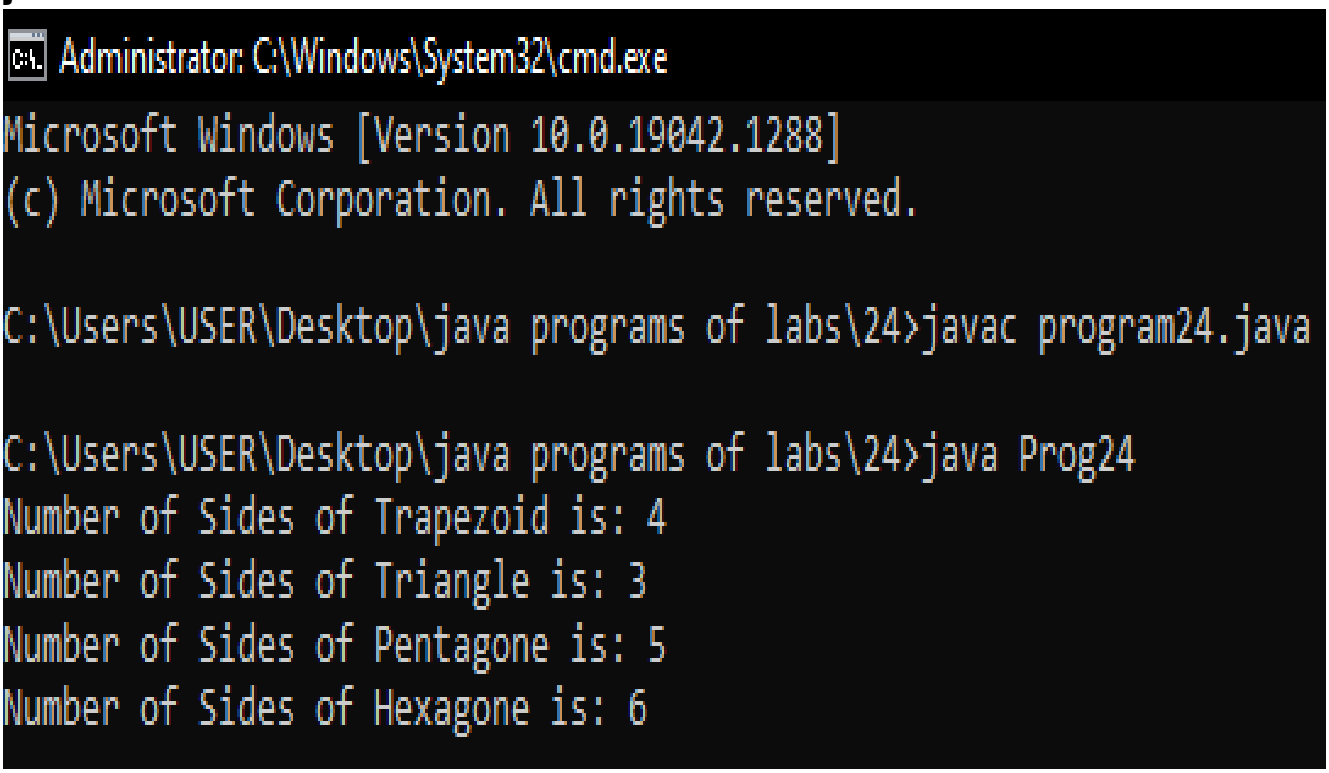


PROGRAM NO:-23

WAP to abstract class name shape that contains an empty method named no of size method.
Provide 4 classes named trapezoid, triangle, pentagon and hexagon such that it overrides the method name of sides and displays the no. of sides in given geometrical figure.

```
abstract class Shape{
   abstract void numberOfSides();
}

class Trapezoid extends Shape{
   public void numberOfSides(){
      System.out.println("Number of Sides of Trapezoid is: "+4);
   }
}

class Triangle extends Shape{
   public void numberOfSides(){
      System.out.println("Number of Sides of Triangle is: "+3);
   }
}

class Pentagon extends Shape{
   public void numberOfSides(){
      System.out.println("Number of Sides of Pentagone is: "+5);
   }
}

class Hexagone extends Shape{
   public void numberOfSides(){
      System.out.println("Number of Sides of Hexagone is: "+6);
   }
}


 class Prog24 {
   public static void main(String[] args) {
   Trapezoid Trap = new Trapezoid();
```

```
    Triangle Tri = new Triangle();
    Pentagon p1 = new Pentagon();
    Hexagone Hex = new Hexagone();

    Trap.numberOfSides();
    Tri.numberOfSides();
    p1.numberOfSides();
    Hex.numberOfSides();
    }
}
```

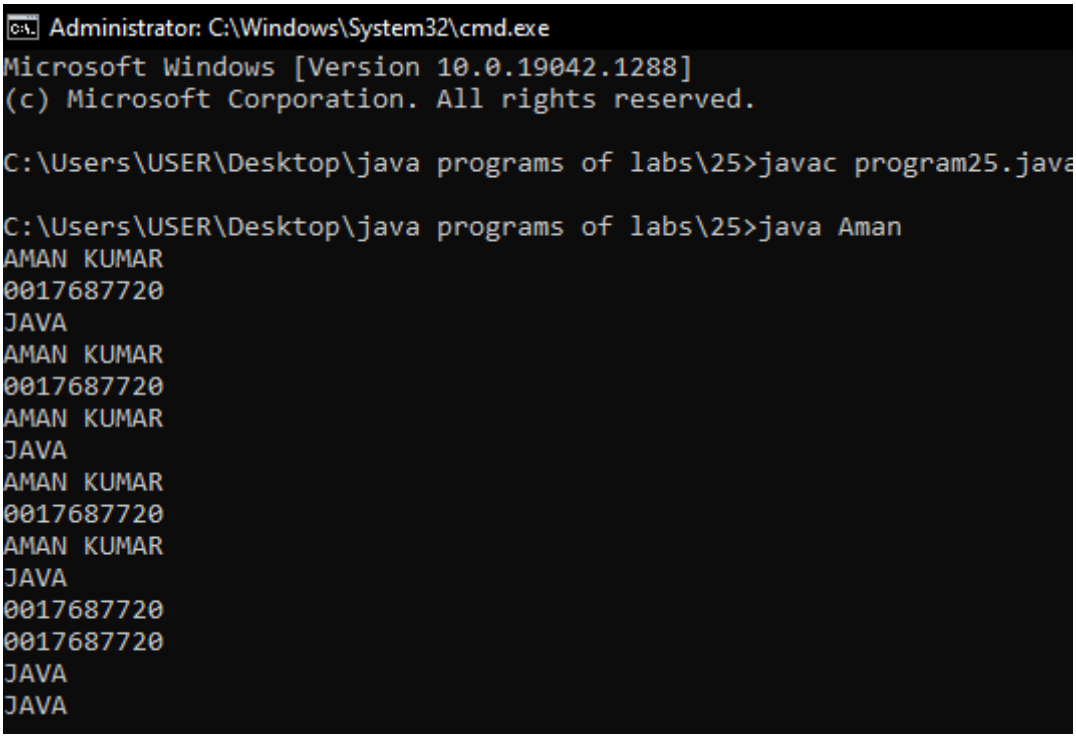


PROGRAM NO:-24

WAP to create three threads by extending thread class. First thread displays your name every 1 second, second thread displays your enrollment number every 2 second and third thread displays your favorite subject in every 3 seconds.

```
class MyThread1 extends Thread {
public void run() {
  int i;
                    try       {
                    for(i=0;i<5;i++){
                              System.out.println("AMAN KUMAR");
                              sleep(1000);
                    }
                    }
                 catch(Exception e)     {
               System.out.println(e);
                    }
                   }
}
class MyThread2 extends Thread{
                  public void run() {
                    int i;
                    try{
                    for(i=0;i<5;i++){
                              System.out.println("0017687720");
                              sleep(2000);
                    }
                    }
                    catch(Exception e){
                              System.out.println(e);
                    }

                   }

}

class MyThread3 extends Thread {

                   public void run() {

                    int i;
```

```
                try

                {for(i=0;i<5;i++){
                        System.out.println("JAVA");
                        sleep(3000);
                }
}
                catch(Exception e)      {
                        System.out.println(e);
                }}
}
class Aman{

               public static void main(String args[]){

                MyThread1 ob1 = new MyThread1();

                MyThread2 ob2 = new MyThread2();

                MyThread3 ob3 = new MyThread3();

                ob1.start();

                ob2.start();

                ob3.start();

               }

}
```

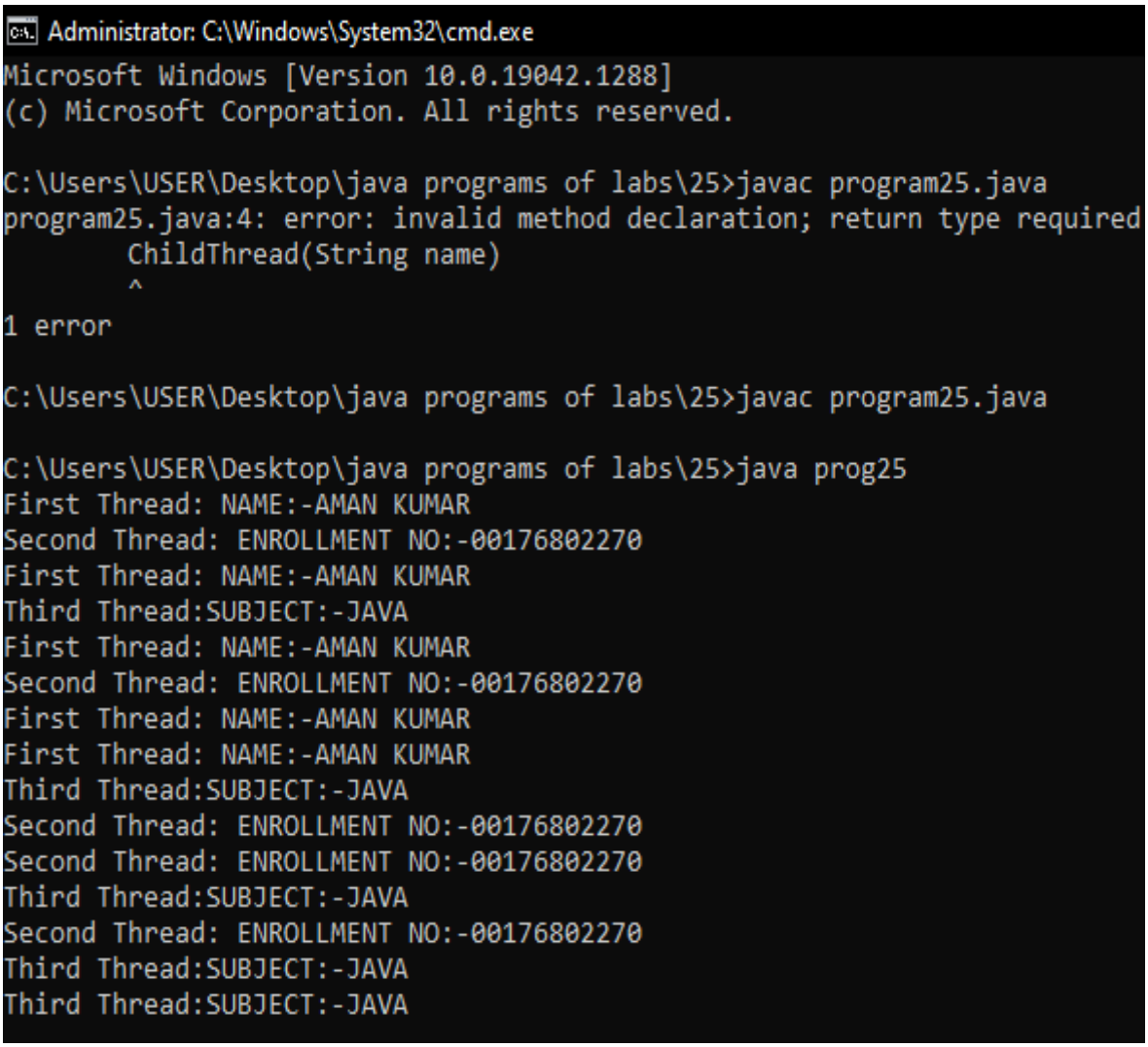

Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1288]
(c) Microsoft Corporation. All rights reserved.
C:\Users\USER\Desktop\java programs of labs\25>javac program25.java
C:\Users\USER\Desktop\java programs of labs\25>java Aman
AMAN KUMAR
0017687720
JAVA
AMAN KUMAR
0017687720
AMAN KUMAR
JAVA
AMAN KUMAR
0017687720
AMAN KUMAR
JAVA
0017687720
0017687720
JAVA
JAVA

PROGRAM NO:-25

First thread displays your name every 1 second, second thread displays your enrollment number every 2 second and third thread displays your favorite subject in every 3 seconds by implementing runnable interface.

```
class ChildThread implements Runnable
{
Thread t;
ChildThread(String name)
{
	t = new Thread(this, name);
	t.start();
}
public void run()
{
	for(int i=1;i<=5;i++)
	{
	try
	{
		if(t.getName().equals("First Thread"))
		{
			Thread.sleep(1000);
			System.out.println(t.getName()+": NAME:-AMAN KUMAR");
		}
		else if(t.getName().equals("Second Thread"))
		{
			Thread.sleep(2000);
			System.out.println(t.getName()+": ENROLLMENT NO:-00176802270");
		}
		else
		{
			Thread.sleep(3000);
			System.out.println(t.getName()+":SUBJECT:-JAVA ");
		}
	}
	catch(InterruptedException e)
	{
		System.out.println(t.getName()+" is interrupted");
	}
	}
}
}
class prog25
{
public static void main(String args[])
{
	ChildThread one = new ChildThread("First Thread");
```

```
        ChildThread two = new ChildThread("Second Thread");
        ChildThread three = new ChildThread("Third Thread");
}
}
```



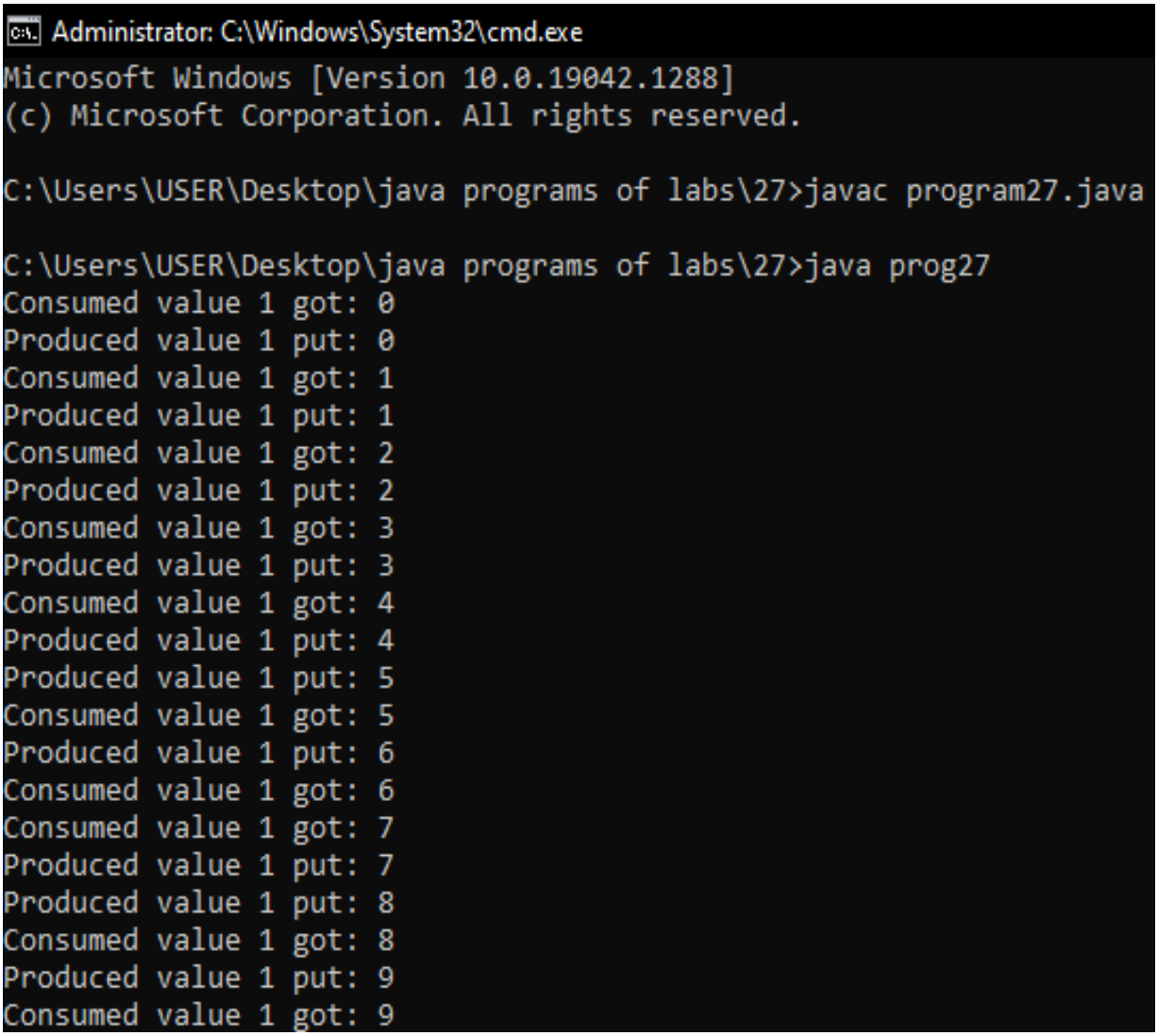
PROGRAM NO:-26

Write a program program of producer and consumer.

```
class prog27
{
    public static void main(String[] args)
    {
        Shop c = new Shop();
        Producer p1 = new Producer(c, 1);
        Consumer c1 = new Consumer(c, 1);
        p1.start();
        c1.start();
    }
}
class Shop
{
    private int materials;
    private boolean available = false;
    public synchronized int get()
    {
        while (available == false)
        {
            try
            {
                wait();
            }
            catch (InterruptedException ie)
            {
            }
        }
        available = false;
        notifyAll();
        return materials;
    }
```

```
    public synchronized void put(int value)
    {
        while (available == true)
        {
            try
            {
                wait();
            }
            catch (InterruptedException ie)
            {
                ie.printStackTrace();
            }
        }
        materials = value;
        available = true;
        notifyAll();
    }
}
class Consumer extends Thread
{
    private Shop Shop;
    private int number;
    public Consumer(Shop c, int number)
    {
        Shop = c;
        this.number = number;
    }
    public void run()
    {
        int value = 0;
        for (int i = 0; i < 10; i++)
        {
            value = Shop.get();
            System.out.println("Consumed value " + this.number+ " got: " +
value);
```

```
        }
    }
}
class Producer extends Thread
{
    private Shop Shop;
    private int number;

    public Producer(Shop c, int number)
    {
        Shop = c;
        this.number = number;
    }
    public void run()
    {
        for (int i = 0; i < 10; i++)
        {
            Shop.put(i);
            System.out.println("Produced value " + this.number+ " put: " + i);
            try
            {
                sleep((int)(Math.random() * 100));
            }
            catch (InterruptedException ie)
            {
                ie.printStackTrace();
            }
        }
    }
}
```


Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1288]
(c) Microsoft Corporation. All rights reserved.
C:\Users\USER\Desktop\java programs of labs\27>javac program27.java
C:\Users\USER\Desktop\java programs of labs\27>java prog27
Consumed value 1 got: 0
Produced value 1 put: 0
Consumed value 1 got: 1
Produced value 1 put: 1
Consumed value 1 got: 2
Produced value 1 put: 2
Consumed value 1 got: 3
Produced value 1 put: 3
Consumed value 1 got: 4
Produced value 1 put: 4
Produced value 1 put: 5
Consumed value 1 got: 5
Produced value 1 put: 6
Consumed value 1 got: 6
Consumed value 1 got: 7
Produced value 1 put: 7
Produced value 1 put: 8
Consumed value 1 got: 8
Produced value 1 put: 9
Consumed value 1 got: 9

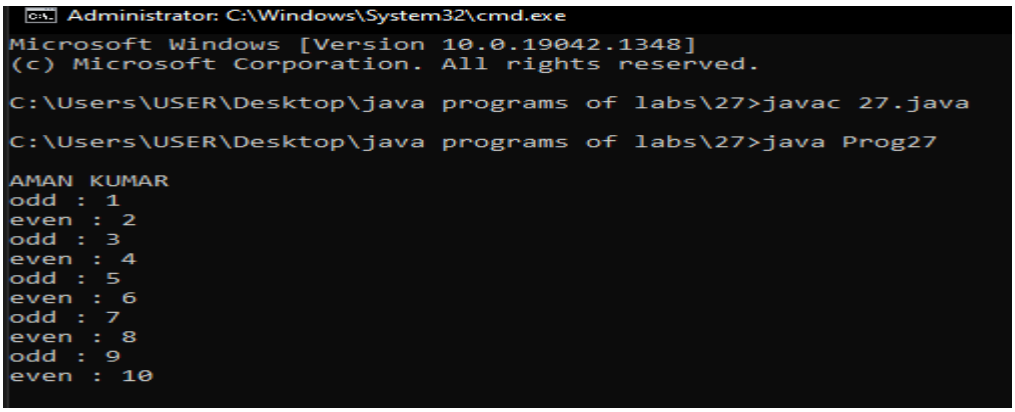
PROGRAM NO:-27

WAP to create two threads one thread displays odd and another thread displays even number.

```
class Prog27 {
public static void main(String[] args) {

Printer printer = new Printer();

MyRunnable r1 = new MyRunnable(true, printer);// isOdd = true
Thread t1 = new Thread(r1);
MyRunnable r2 = new MyRunnable(false, printer);// isOdd = false
Thread t2 = new Thread(r2);
t1.start();
t2.start();

System.out.println(" ");
System.out.println("AMAN KUMAR");
}
}

class Printer {
private Object lock = new Object();
private volatile boolean isOdd = false;

public void printEven(int number) throws InterruptedException {
synchronized (lock) {
while (isOdd == false) {
lock.wait();
}
System.out.println("even : " + number);
isOdd = false;
lock.notifyAll();
}
}

public void printOdd(int number) throws InterruptedException {
synchronized (lock) {
while (isOdd == true) {
lock.wait();
}
System.out.println("odd : " + number);
isOdd = true;
lock.notifyAll();
```

```
}
}
}

class MyRunnable implements Runnable {

private boolean isOdd;
Printer printer;

MyRunnable(boolean isOdd, Printer printer) {
this.isOdd = isOdd;
this.printer = printer;
}

public void run() {
int number = isOdd == true ? 1 : 2;
while (number <= 10) {
if (isOdd) {
try {
printer.printOdd(number);
} catch (InterruptedException e) {
}
} else {
try {
printer.printEven(number);
} catch (InterruptedException e) {
}
}
number += 2;
}
}
}
```

OUTPUT:-



PROGRAM NO:-28

Write an applet that displays a simple message in a different form and also set the background color and foreground color

```
import java.awt.*;
import java.applet.*;

/*
<applet code="Prog2" width=300 height=300></applet>
*/


public class Prog2 extends Applet {
   Font f1, f2, f3;
   public void init() {
      setBackground(Color.cyan);
      setForeground(Color.orange);
      f1 = new Font("Elephant", Font.ITALIC, 50);
      f2 = new Font("Arial", Font.BOLD, 40);
      f3 = new Font("Forte", Font.PLAIN, 30);
   }

   public void paint(Graphics g) {

      g.setFont(f1);
      g.drawString("AMAN KUMAR", 50, 100);

      g.setColor(Color.PINK);
      g.setFont(f2);
      g.drawString("SHIN-CHAN", 50, 150);

      g.setColor(Color.GREEN);
      g.setFont(f3);
      g.drawString("PROFESSIONAL ", 50, 200);

   }
}
```

OUTPUT:-


Applet Viewer: Prog2
Applet
.A.M.A.N KU.M.AR
SHIN-CHAN
PROFESSIONAL
Applet started.

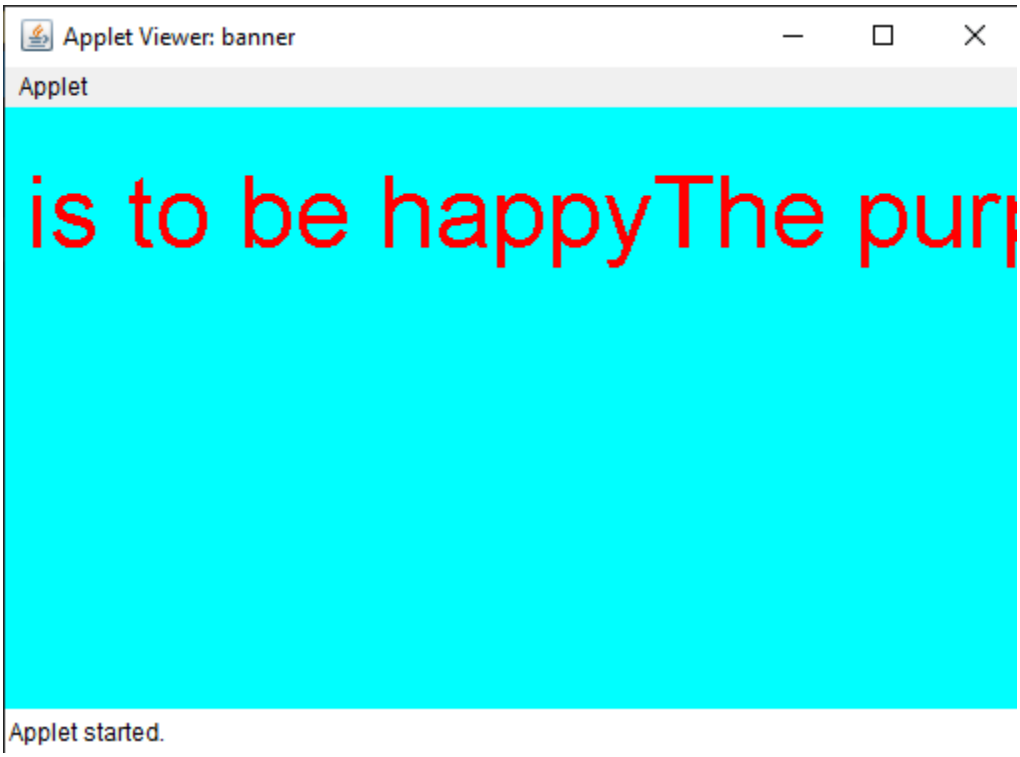
PROGRAM NO:-29

Write an applet that displays a moving banner from left coordinate to right coordinate in a circular way.

```
import java.awt.*;
import java.applet.*;

/*
<applet code="banner" width=300 height=300>
<param name="msg" value="The purpose of our lives is to be happy">
</applet>
*/

public class banner extends Applet implements Runnable {

   String banner = "The purpose of our lives is to be happy";
   Thread t = null;
   int state;
   boolean flag;
   Font f1;

   public void init() {
      setBackground(Color.cyan);
      setForeground(Color.red);
      f1 = new Font("Arial", Font.PLAIN, 50);
   }

   public void start() {
      t = new Thread(this);
      flag = false;
      t.start();
   }

   public void run() {
      while(true) {
         try {
            repaint();
            Thread.sleep(250);
            char ch = banner.charAt(0);
            banner = banner.substring(1, banner.length());
            banner += ch;
            if (flag)
               break;
         } catch (InterruptedException e) {}
      }
   }

   public void stop() {
```

```
        flag = true;
        t = null;
    }

    public void paint(Graphics g) {
        g.setFont(f1);
        g.drawString(banner, 10, 70);
    }
}
```

OUTPUT:-



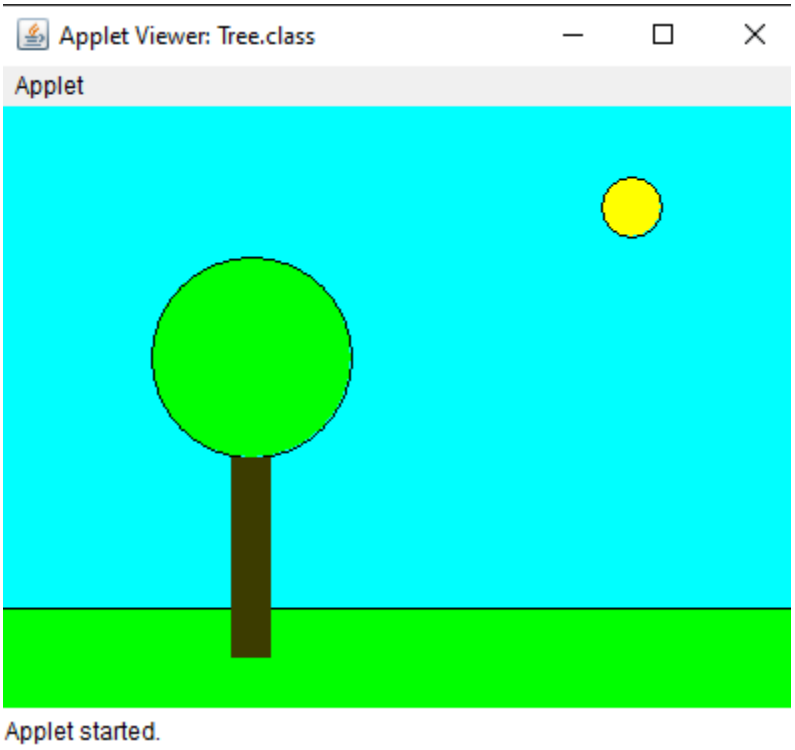
PROGRAM NO:-30

Write an applet to draw a scenery using graphics class.

```
import java.applet.Applet;
import java.awt.*;

public class Tree extends Applet
{
    public void init()
    {
        setSize(400, 300);
        setBackground(Color.cyan);
    }

    public void paint(Graphics g)
    {
        // Draw grass.
        g.setColor(Color.green);
        g.fillRect(0, 250, 400, 50);
        g.setColor(Color.black);
        g.drawLine(0, 250, 400, 250);

        // Draw leaves of tree.
        g.setColor(Color.green);
        g.fillOval(75, 75, 100, 100);
        g.setColor(Color.black);
        g.drawOval(75, 75, 100, 100);

        // Draw sun.
        g.setColor(Color.yellow);
        g.fillOval(300, 35, 30, 30);
        g.setColor(Color.black);
        g.drawOval(300, 35, 30, 30);

        // Draw trunk of tree.
        g.setColor(new Color(60, 60, 0));
        g.fillRect(115, 175, 20, 100);
```

```
    }
}

/*
  <applet code="Tree.class" width="300" height="400">
</applet>
*/
```

OUTPUT:-



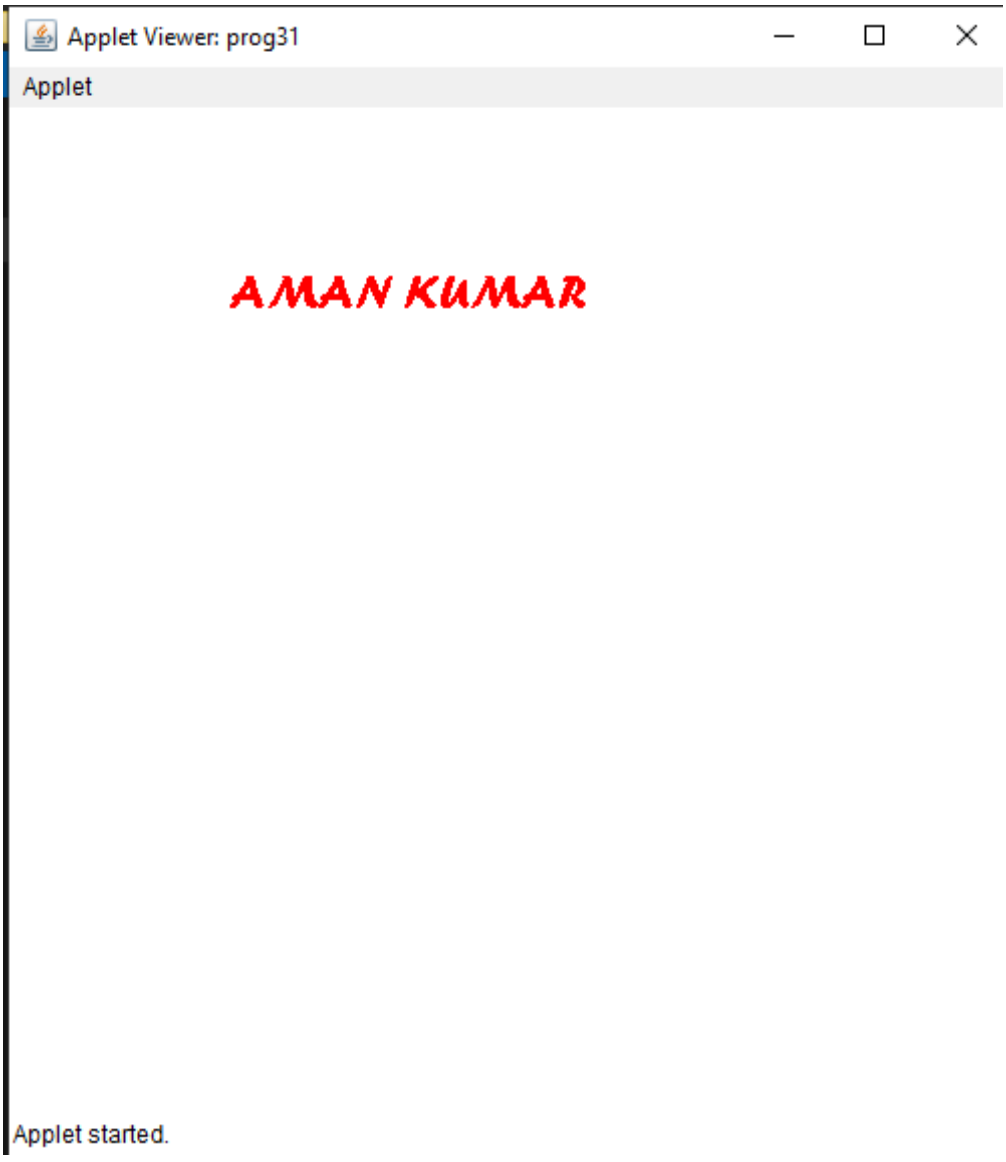
PROGRAM NO:-31

Write a applet that displays a moving banner with the text provided by the parameter tag

```
/*<APPLET code = "prog31" width = 500 height = 500 >
<param name="msg" value="AMAN KUMAR">
</APPLET>
*/

import java.awt.*;
import java.applet.*;

public class prog31 extends Applet implements Runnable {
private String display;
private int x, y, flag;
Thread t;
Font f2;
public void init()
{
	f2 = new Font("Forte",Font.PLAIN,24);

	display = getParameter("msg");
	x = 500;
	y = 100;
	flag = 1;

	t = new Thread(this, "MyThread");

	t.start();
}

public void update()
{
	x = x + 10 * flag;
	if (x > 300)
		flag = -1;
	if (x < 100)
		flag = +1;
}

public void run()
{
	while (true) {
		repaint();
		update();
		try {
			Thread.sleep(250);
		}
```

```
                catch (InterruptedException ie) {
                        System.out.println(ie);
                }
        }
}

public void paint(Graphics g)
{
        setForeground(Color.red);
        g.setFont(f2);
        g.drawString(display, x, y);
}
}
```

OUTPUT:-



PROGRAM NO:-32

Write an applet that displays the usage of parameter tags using various data types

```
/*<APPLET code = "Prog1" width = 500 height = 500 >
<param name="string" value="AMAN KUMAR">
<param name="roll" value="00176807720">
</APPLET>
*/
import java.awt.*;
import java.applet.*;
public class Prog1 extends Applet {
   private String display;
   private String roll;
   Font f2;
public void init()
{
        f2 = new Font("Forte",Font.BOLD,24);
        display = getParameter("String");
     roll = getParameter("roll");
}
public void paint(Graphics g)
{
        setForeground(Color.red);
        g.setFont(f2);
        g.drawString(display, 100, 100);
        g.drawString(roll, 100, 150);
}
}
```

OUTPUT:-


Applet Viewer: Prog1
Applet
AMAN KUMAR
00176807720
Applet started.

PROGRAM NO:-33

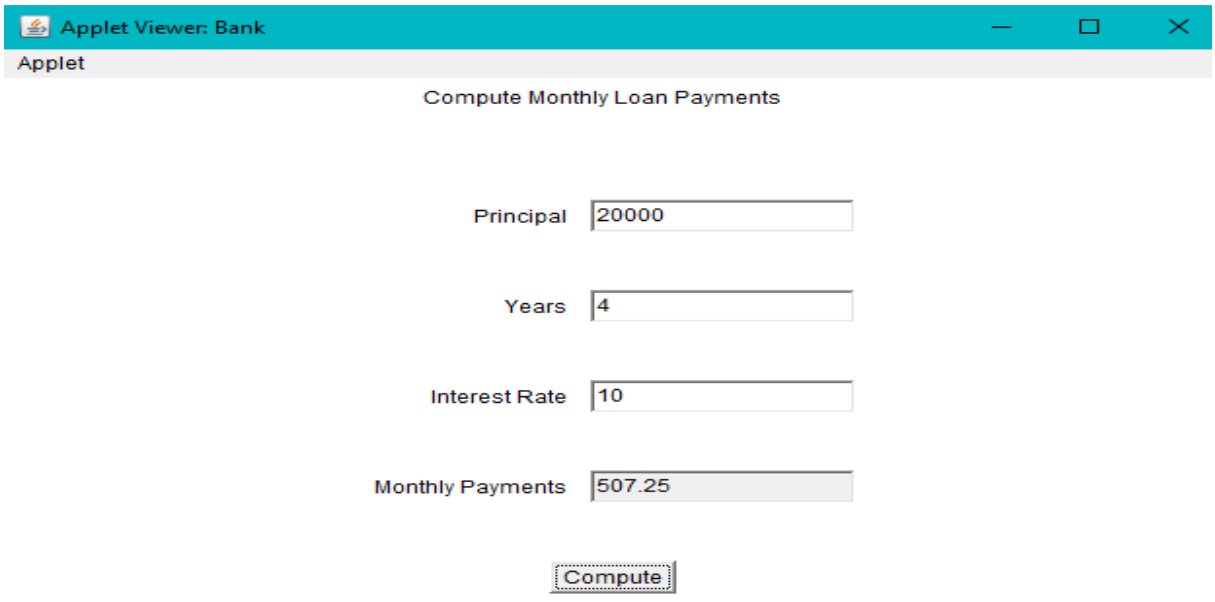
To write an applet that computes the payment of a loan based on the amount of the loan, the interest rate and the number of months. It takes one parameter from the browser monthly rate. If true the interest rate is calculated monthly otherwise the interest rate is calculated annually.

```
import java.applet.Applet;
import java.awt.Button;
import java.awt.Graphics;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.Label;
import java.awt.TextField;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.text.NumberFormat;
/*
<applet code="Bank" width=280 height=200></applet>
*/
public class Bank extends Applet implements ActionListener {
   TextField amountText, paymentText, periodText, rateText;
   Button doIt;
   double principal; // original princial
   double intRate; // interest rate
   double numYears; // length of loan in years
   /*
   * Number of payments per year. You could allow this value to be set by the
   * user.
   */
   final int payPerYear = 12;
   NumberFormat nf;
   public void init() {
      // Use a grid bag layout.
      GridBagLayout gbag = new GridBagLayout();
      GridBagConstraints gbc = new GridBagConstraints();
      setLayout(gbag);
      Label heading = new Label("Compute Monthly Loan Payments");
      Label amountLab = new Label("Principal");
      Label periodLab = new Label("Years");
      Label rateLab = new Label("Interest Rate");
      Label paymentLab = new Label("Monthly Payments");
      amountText = new TextField(16);
      periodText = new TextField(16);
      paymentText = new TextField(16);
      rateText = new TextField(16);
      // Payment field for display only.
      paymentText.setEditable(false);
      doIt = new Button("Compute");
      // Define the grid bag.
      gbc.weighty = 1.0; // use a row weight of 1
```

```
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    gbc.anchor = GridBagConstraints.NORTH;
    gbag.setConstraints(heading, gbc);
    // Anchor most components to the right.
    gbc.anchor = GridBagConstraints.EAST;
    gbc.gridwidth = GridBagConstraints.RELATIVE;
    gbag.setConstraints(amountLab, gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    gbag.setConstraints(amountText, gbc);
    gbc.gridwidth = GridBagConstraints.RELATIVE;
    gbag.setConstraints(periodLab, gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    gbag.setConstraints(periodText, gbc);
    gbc.gridwidth = GridBagConstraints.RELATIVE;
    gbag.setConstraints(rateLab, gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    gbag.setConstraints(rateText, gbc);
    gbc.gridwidth = GridBagConstraints.RELATIVE;
    gbag.setConstraints(paymentLab, gbc);
    gbc.gridwidth = GridBagConstraints.REMAINDER;
    gbag.setConstraints(paymentText, gbc);
    gbc.anchor = GridBagConstraints.CENTER;
    gbag.setConstraints(doIt, gbc);
    // Add all the components.
    add(heading);
    add(amountLab);
    add(amountText);
    add(periodLab);
    add(periodText);
    add(rateLab);
    add(rateText);
    add(paymentLab);
    add(paymentText);
    add(doIt);
    // Register to receive action events.
    amountText.addActionListener(this);
    periodText.addActionListener(this);
    rateText.addActionListener(this);
    doIt.addActionListener(this);
    nf = NumberFormat.getInstance();
    nf.setMinimumFractionDigits(2);
    nf.setMaximumFractionDigits(2);
}
/*
* User pressed Enter on a text field or pressed Compute.
*/
public void actionPerformed(ActionEvent ae) {
    repaint();
}
// Display the result if all fields are completed.
public void paint(Graphics g) {
```

```
        double result = 0.0;
        String amountStr = amountText.getText();
        String periodStr = periodText.getText();
        String rateStr = rateText.getText();
        try {
        if (amountStr.length() != 0 && periodStr.length() != 0
&& rateStr.length() != 0) {
        principal = Double.parseDouble(amountStr);
        numYears = Double.parseDouble(periodStr);
        intRate = Double.parseDouble(rateStr) / 100;
        result = compute();
        paymentText.setText(nf.format(result));
        }
        showStatus(""); // erase any previous error message
        } catch (NumberFormatException exc) {
        showStatus("Invalid Data");
        paymentText.setText("");
        }
        }
        // Compute the loan payment.
        double compute() {
        double numer;
        double denom;
        double b, e;
        numer = intRate * principal / payPerYear;
        e = -(payPerYear * numYears);
        b = (intRate / payPerYear) + 1.0;
        denom = 1.0 - Math.pow(b, e);
        return numer / denom;
    }
}
```

OUTPUT:-



PROGRAM NO:-34

Create a form using the following components in an applet TextField, TextArea, Button, Label.

```
import java.awt.*;

class BasicAWT
{
 public static void main(String args[])
 {
  Frame f = new Frame();
  f.setSize(400,400);
  f.setVisible(true);
  f.setLayout(new FlowLayout() );

  Label l1 = new Label();
  l1.setText("Enter Your Name ");

  TextField tf = new TextField("Aman kumar");

  Label l2 = new Label("Address");
  TextArea ta = new TextArea("",3,40);

  Button b = new Button("Submit");

  f.add(l1); f.add(tf); f.add(l2); f.add(ta); f.add(b);
 }
}
```

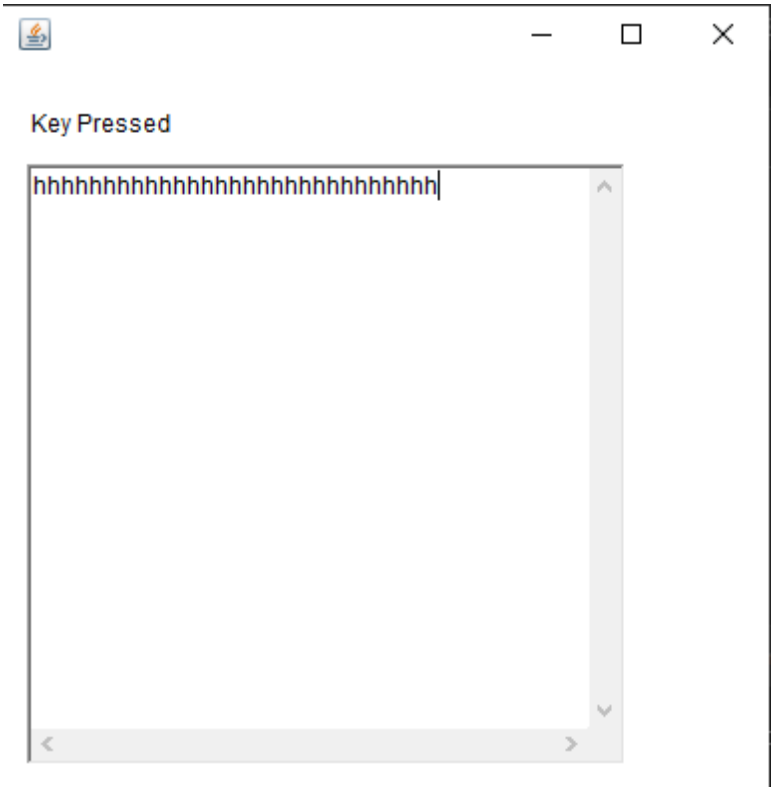
OUTPUT:-

PROGRAM NO:-35

To develop a program to handle keyboard event.

```
import java.awt.*;
import java.awt.event.*;
public class KeyListenerExample extends Frame implements KeyListener
{
 Label l;
 TextArea area;
 KeyListenerExample(){
 l=new Label();
 l.setBounds(20,50,100,20);
 area=new TextArea();
 area.setBounds(20,80,300, 300);
 area.addKeyListener(this);
 add(l);
 add(area);
 setSize(400,400);
 setLayout(null);
 setVisible(true);
 }
 public void keyPressed(KeyEvent e) {
 l.setText("Key Pressed");
 }
 public void keyReleased(KeyEvent e) {
 l.setText("Key Released");
 }
 public void keyTyped(KeyEvent e) {
 l.setText("Key Typed");
 }
 public static void main(String[] args) {
 new KeyListenerExample();
 }
 }
```

OUTPUT:-


Key Pressed
hhhhhhhhhhhhhhhhhhhhhhhhhhhhhh

PROGRAM NO:-36

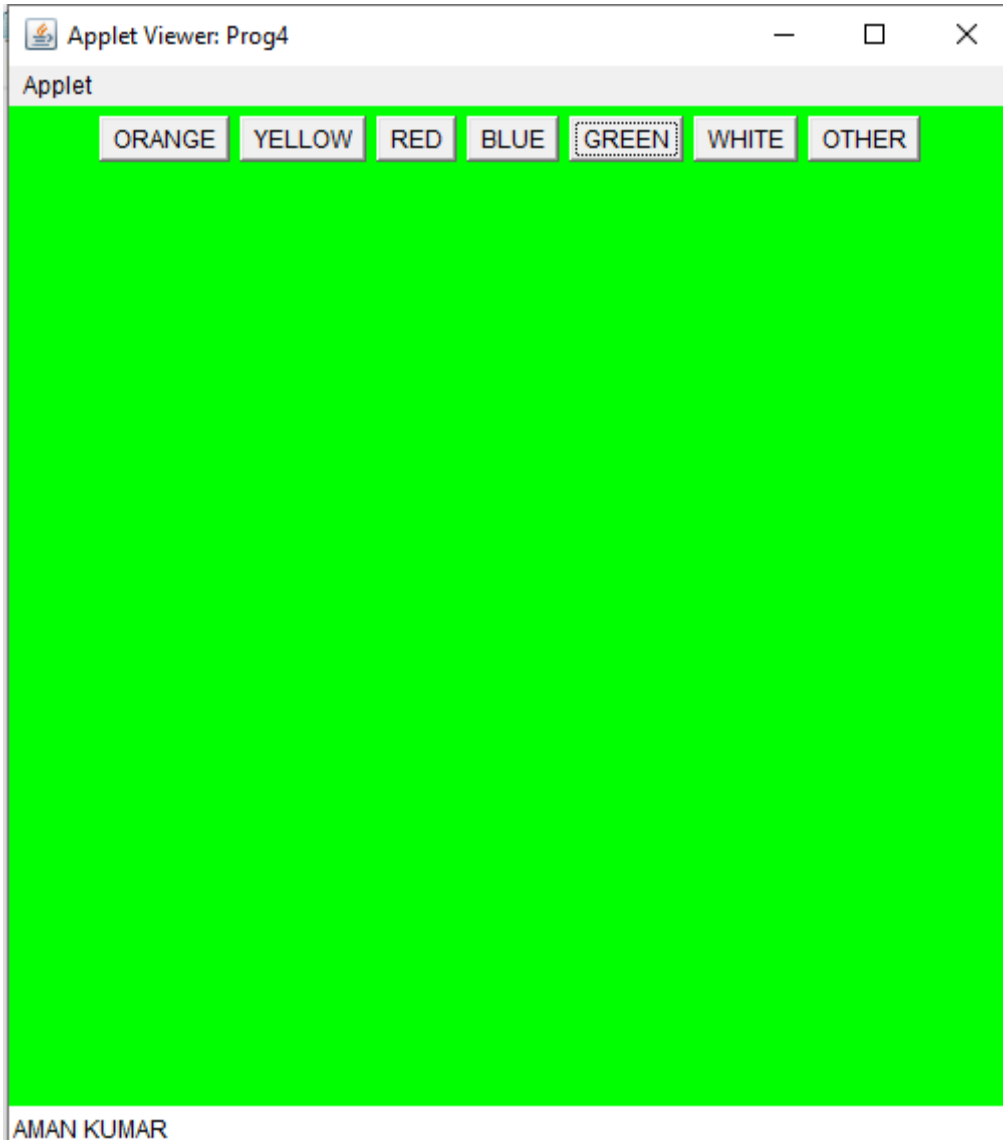
Create an applet that contains seven buttons, each button is for the color of the rainbow. Whenever a button is pressed its corresponding colors should be filled in the background.

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
/*<APPLET code = "Prog4" width = 500 height = 500 >
<param name="msg" value="AMAN KUMAR">
</APPLET>ss
*/

public class Prog4 extends Applet implements ActionListener{
      Button b1;
      Button b2;
      Button b3;
      Button b4;
      Button b5;
      Button b6;
      Button b7;
   public void init(){



      b1 = new Button("ORANGE");
      b2 = new Button("YELLOW");
      b3 = new Button("RED");
      b4 = new Button("BLUE");
      b5 = new Button("GREEN");
      b6 = new Button("WHITE");
      b7 = new Button("OTHER");

      b1.addActionListener(this);
      b2.addActionListener(this);
      b3.addActionListener(this);
      b4.addActionListener(this);
      b5.addActionListener(this);
      b6.addActionListener(this);
      b7.addActionListener(this);
```

```
    add(b1);
    add(b2);
    add(b3);
    add(b4);
    add(b5);
    add(b6);
    add(b7);
}
public void actionPerformed(ActionEvent e){
    if(e.getSource()==b1){

        setBackground(Color.orange);
    }
    else if(e.getSource()==b2){

        setBackground(Color.yellow);
    }
    else if(e.getSource()==b3){

        setBackground(Color.red);
    }
    else if(e.getSource()==b4){

        setBackground(Color.blue);
    }
    else if(e.getSource()==b5){

        setBackground(Color.green);
    }
    else if(e.getSource()==b6){

        setBackground(Color.white);
    }
    else if(e.getSource()==b7){

        setBackground(Color.black);
    }


}
```

```
    public void paint(Graphics g)
    {
        showStatus("AMAN KUMAR");
    }
}
```

OUTPUT:-



PROGRAM NO:-37

Create a text editor like MS WORD

```
import java.awt.*;
import javax.swing.*;
import java.io.*;
import java.awt.event.*;
import javax.swing.plaf.metal.*;
import javax.swing.text.*;
class editor extends JFrame implements ActionListener {
// Text component
JTextArea t;

// Frame
JFrame f;

// Constructor
editor()
{
	// Create a frame
	f = new JFrame("editor");

	try {
		// Set metal look and feel

	UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLo
okAndFeel");

		// Set theme to ocean
		MetalLookAndFeel.setCurrentTheme(new
OceanTheme());
	}
```

```
catch (Exception e) {
}

// Text component
t = new JTextArea();

// Create a menubar
JMenuBar mb = new JMenuBar();

// Create amenu for menu
JMenu m1 = new JMenu("File");

// Create menu items
JMenuItem mi1 = new JMenuItem("New");
JMenuItem mi2 = new JMenuItem("Open");
JMenuItem mi3 = new JMenuItem("Save");
JMenuItem mi9 = new JMenuItem("Print");

// Add action listener
mi1.addActionListener(this);
mi2.addActionListener(this);
mi3.addActionListener(this);
mi9.addActionListener(this);

m1.add(mi1);
m1.add(mi2);
m1.add(mi3);
m1.add(mi9);

// Create amenu for menu
JMenu m2 = new JMenu("Edit");

// Create menu items
```

```
        JMenuItem mi4 = new JMenuItem("cut");
        JMenuItem mi5 = new JMenuItem("copy");
        JMenuItem mi6 = new JMenuItem("paste");

        // Add action listener
        mi4.addActionListener(this);
        mi5.addActionListener(this);
        mi6.addActionListener(this);

        m2.add(mi4);
        m2.add(mi5);
        m2.add(mi6);

        JMenuItem mc = new JMenuItem("close");

        mc.addActionListener(this);

        mb.add(m1);
        mb.add(m2);
        mb.add(mc);

        f.setJMenuBar(mb);
        f.add(t);
        f.setSize(500, 500);
        f.show();
}

// If a button is pressed
public void actionPerformed(ActionEvent e)
{
        String s = e.getActionCommand();

        if (s.equals("cut")) {
```

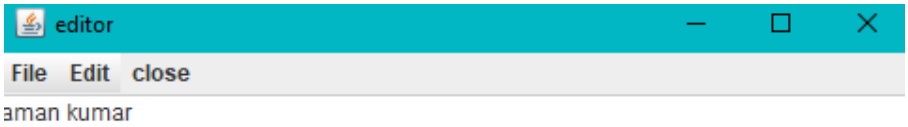
```
            t.cut();
        }
        else if (s.equals("copy")) {
            t.copy();
        }
        else if (s.equals("paste")) {
            t.paste();
        }
        else if (s.equals("Save")) {
            // Create an object of JFileChooser class
            JFileChooser j = new JFileChooser("f:");

            // Invoke the showsSaveDialog function to show the
save dialog
            int r = j.showSaveDialog(null);

            if (r == JFileChooser.APPROVE_OPTION) {

                // Set the label to the path of the selected directory
                File fi = new
File(j.getSelectedFile().getAbsolutePath());

                try {
                    // Create a file writer
                    FileWriter wr = new FileWriter(fi, false);

                    // Create buffered writer to write
                    BufferedWriter w = new BufferedWriter(wr);

                    // Write
                    w.write(t.getText());

                    w.flush();
```

```
                    w.close();
                }
                catch (Exception evt) {
                    JOptionPane.showMessageDialog(f,
evt.getMessage());
                }
            }
            // If the user cancelled the operation
            else
                JOptionPane.showMessageDialog(f, "the user
cancelled the operation");
        }
        else if (s.equals("Print")) {
            try {
                // print the file
                t.print();
            }
            catch (Exception evt) {
                JOptionPane.showMessageDialog(f,
evt.getMessage());
            }
        }
        else if (s.equals("Open")) {
            // Create an object of JFileChooser class
            JFileChooser j = new JFileChooser("f:");

            // Invoke the showsOpenDialog function to show the
save dialog
            int r = j.showOpenDialog(null);

            // If the user selects a file
            if (r == JFileChooser.APPROVE_OPTION) {
                // Set the label to the path of the selected directory
```

```
                File fi = new
File(j.getSelectedFile().getAbsolutePath());

                try {
                        // String
                        String s1 = "", sl = "";

                        // File reader
                        FileReader fr = new FileReader(fi);

                        // Buffered reader
                        BufferedReader br = new BufferedReader(fr);

                        // Initialize sl
                        sl = br.readLine();

                        // Take the input from the file
                        while ((s1 = br.readLine()) != null) {
                                sl = sl + "\n" + s1;
                        }

                        // Set the text
                        t.setText(sl);
                }
                catch (Exception evt) {
                        JOptionPane.showMessageDialog(f,
evt.getMessage());
                }
        }
        // If the user cancelled the operation
        else
                JOptionPane.showMessageDialog(f, "the user
cancelled the operation");
```

```
        }
        else if (s.equals("New")) {
                t.setText("");
        }
        else if (s.equals("close")) {
                f.setVisible(false);
        }
}

// Main class
public static void main(String args[])
{
        editor e = new editor();
}
}
```

OUTPUT:-



PROGRAM NO:-38

Make an analog Clock

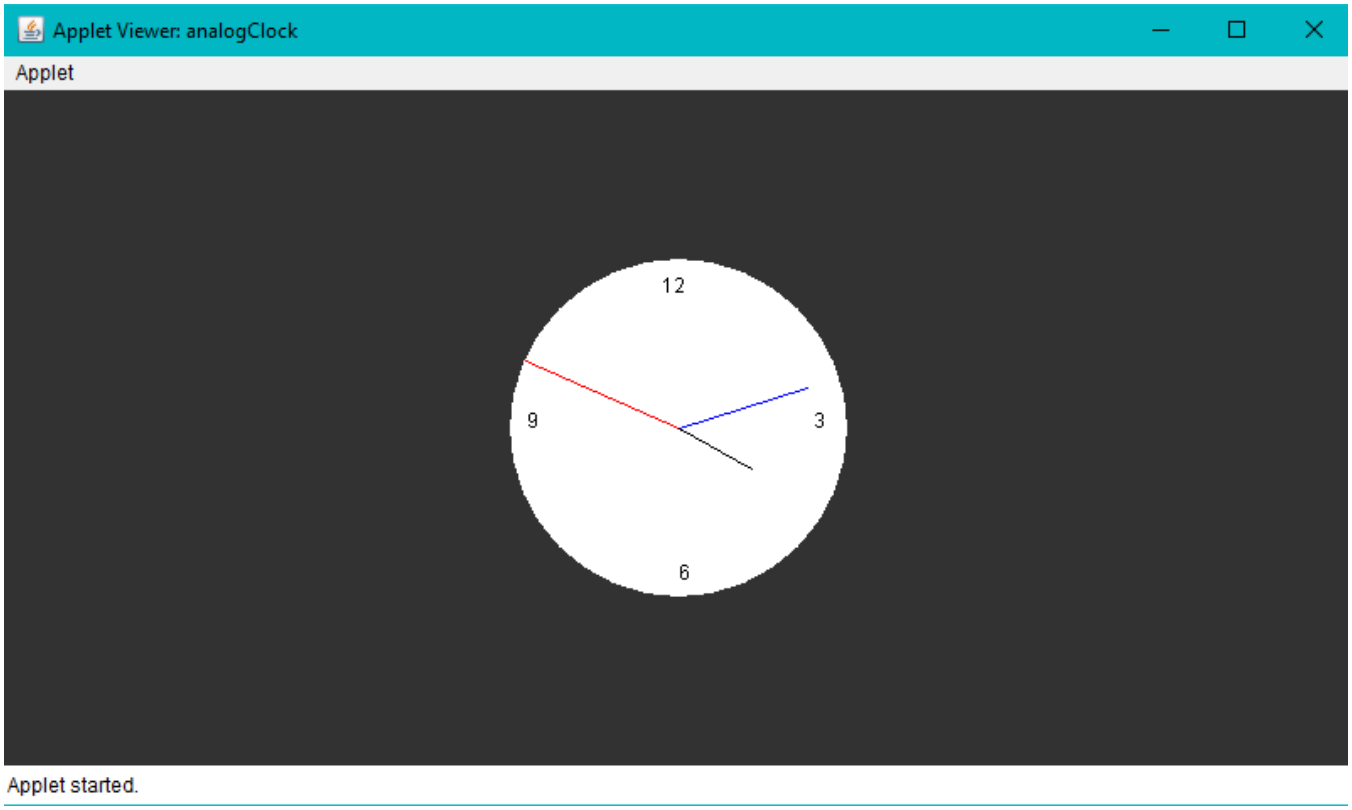
```
import java.applet.Applet;
import java.awt.*;
import java.util.*;
/*<APPLET code = "analogClock" width = 500 height = 500 >
<param name="msg" value="AMAN KUMAR">
</APPLET>ss
*/


public class analogClock extends Applet {

@Override
public void init()
{
	this.setSize(new Dimension(800, 400));
	setBackground(new Color(50, 50, 50));
	new Thread() {
		@Override
		public void run()
		{
			while (true) {
				repaint();
				delayAnimation();
			}
		}
	}.start();
}
```

```
private void delayAnimation()
{
	try {

		Thread.sleep(1000);
	}
	catch (InterruptedException e) {
		e.printStackTrace();
	}
}

@Override
public void paint(Graphics g)
{

	Calendar time = Calendar.getInstance();

	int hour = time.get(Calendar.HOUR_OF_DAY);
	int minute = time.get(Calendar.MINUTE);
	int second = time.get(Calendar.SECOND);



	if (hour > 12) {
		hour -= 12;
	}



	g.setColor(Color.white);
	g.fillOval(300, 100, 200, 200);



	g.setColor(Color.black);
	g.drawString("12", 390, 120);
	g.drawString("9", 310, 200);
```

```
g.drawString("6", 400, 290);
g.drawString("3", 480, 200);


double angle;
int x, y;


angle = Math.toRadians((15 - second) * 6);



x = (int)(Math.cos(angle) * 100);
y = (int)(Math.sin(angle) * 100);


g.setColor(Color.red);
g.drawLine(400, 200, 400 + x, 200 - y);


angle = Math.toRadians((15 - minute) * 6);


x = (int)(Math.cos(angle) * 80);
y = (int)(Math.sin(angle) * 80);

g.setColor(Color.blue);
g.drawLine(400, 200, 400 + x, 200 - y);

angle = Math.toRadians((15 - (hour * 5)) * 6);


x = (int)(Math.cos(angle) * 50);
y = (int)(Math.sin(angle) * 50);
```

```
        g.setColor(Color.black);
        g.drawLine(400, 200, 400 + x, 200 - y);
}
}
```

OUTPUT:-



PROGRAM NO:-39

Make a Normal calculator

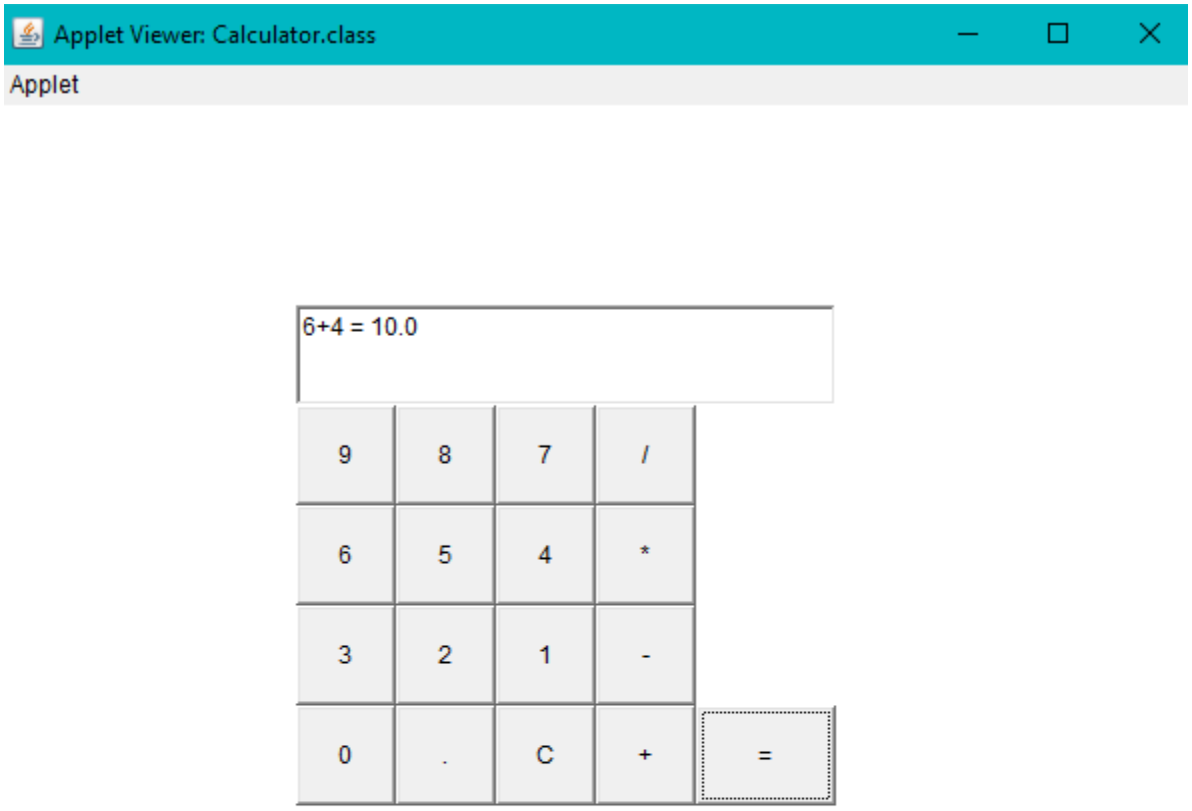
```
/*Java Program to Demonstrate a Basic Calculator using Applet*/
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class Calculator extends Applet implements ActionListener
{
   TextField inp;
   //Function to add features to the frame
   public void init()
   {
setBackground(Color.white);
setLayout(null);
int i;
inp = new TextField();
inp.setBounds(150,100,270,50);
this.add(inp);
Button button[] = new Button[10];
for(i=0;i<10;i++)
{
   button[i] = new Button(String.valueOf(9-i));
   button[i].setBounds(150+((i%3)*50),150+((i/3)*50),50,50);
   this.add(button[i]);
   button[i].addActionListener(this);
}
Button dec=new Button(".");
dec.setBounds(200,300,50,50);
this.add(dec);
dec.addActionListener(this);

Button clr=new Button("C");
clr.setBounds(250,300,50,50);
this.add(clr);
clr.addActionListener(this);
```

```
Button operator[] = new Button[5];
operator[0]=new Button("/");
operator[1]=new Button("*");
operator[2]=new Button("-");
operator[3]=new Button("+");
operator[4]=new Button("=");
for(i=0;i<4;i++)
{
   operator[i].setBounds(300,150+(i*50),50,50);
   this.add(operator[i]);
   operator[i].addActionListener(this);
}
operator[4].setBounds(350,300,70,50);
this.add(operator[4]);
operator[4].addActionListener(this);
   }
   String num1="";
   String op="";
   String num2="";
   //Function to calculate the expression
   public void actionPerformed(ActionEvent e)
   {
String button = e.getActionCommand();
      char ch = button.charAt(0);
if(ch>='0' && ch<='9'|| ch=='.')
{
   if (!op.equals(""))
         num2 = num2 + button;
   else
         num1 = num1 + button;
   inp.setText(num1+op+num2);
}
else if(ch=='C')
{
   num1 = op = num2 = "";
   inp.setText("");
}
else if (ch =='=')
{
   if(!num1.equals("") && !num2.equals(""))
```

```
   {
        double temp;
        double n1=Double.parseDouble(num1);
        double n2=Double.parseDouble(num2);
        if(n2==0 && op.equals("/"))
        {
           inp.setText(num1+op+num2+" = Zero Division Error");
           num1 = op = num2 = "";
        }
        else
        {
           if (op.equals("+"))
              temp = n1 + n2;
           else if (op.equals("-"))
              temp = n1 - n2;
           else if (op.equals("/"))
              temp = n1/n2;
           else
              temp = n1*n2;
           inp.setText(num1+op+num2+" = "+temp);
           num1 = Double.toString(temp);
           op = num2 = "";
      }
         }
   else
   {
        num1 = op = num2 = "";
        inp.setText("");
   }
     }
else
{
   if (op.equals("") || num2.equals(""))
        op = button;
   else
   {
        double temp;
        double n1=Double.parseDouble(num1);
        double n2=Double.parseDouble(num2);
        if(n2==0 && op.equals("/"))
```

```
        {
           inp.setText(num1+op+num2+" = Zero Division Error");
           num1 = op = num2 = "";
        }
        else
        {
           if (op.equals("+"))
              temp = n1 + n2;
           else if (op.equals("-"))
              temp = n1 - n2;
           else if (op.equals("/"))
              temp = n1/n2;
           else
              temp = n1*n2;
           num1 = Double.toString(temp);
           op = button;
           num2 = "";
      }
         }
   inp.setText(num1+op+num2);
      }
   }
}
/*
<applet code = Calculator.class width=600 height=600>
</applet>
*/
```

OUTPUT:-



Applet started.

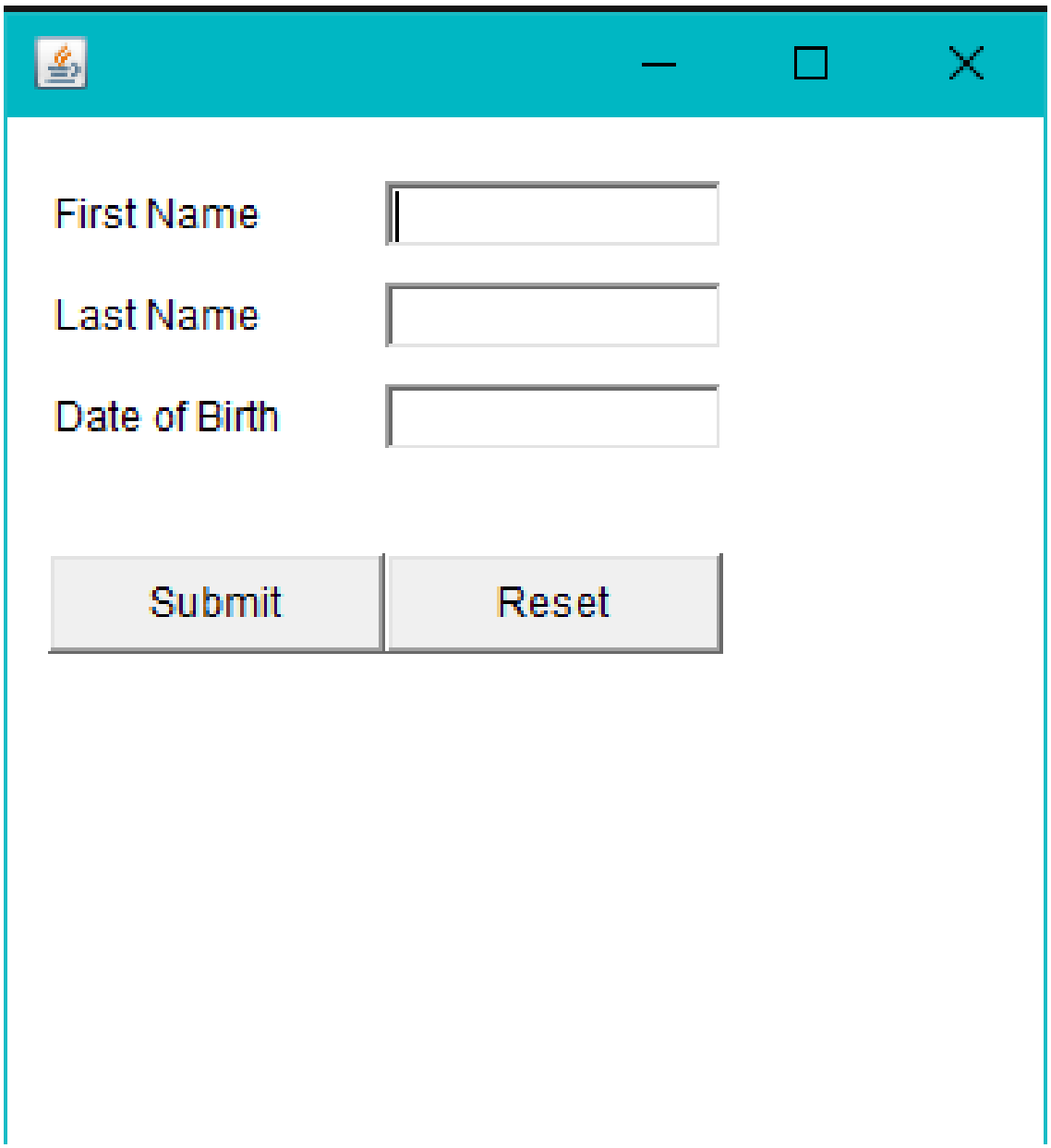
PROGRAM NO:-40

To develop a program to create an extended awt component.

```
import java.awt.*;
 class Aman extends Frame {

Aman(){
Label firstName = new Label("First Name");
firstName.setBounds(20, 50, 80, 20);

Label lastName = new Label("Last Name");
lastName.setBounds(20, 80, 80, 20);

Label dob = new Label("Date of Birth");
dob.setBounds(20, 110, 80, 20);

TextField firstNameTF = new TextField();
firstNameTF.setBounds(120, 50, 100, 20);

TextField lastNameTF = new TextField();
lastNameTF.setBounds(120, 80, 100, 20);

TextField dobTF = new TextField();
dobTF.setBounds(120, 110, 100, 20);

Button sbmt = new Button("Submit");
sbmt.setBounds(20, 160, 100, 30);

Button reset = new Button("Reset");
reset.setBounds(120,160,100,30);

add(firstName);
add(lastName);
add(dob);
add(firstNameTF);
add(lastNameTF);
add(dobTF);
add(sbmt);
```

```
add(reset);

setSize(300,300);
setLayout(null);
setVisible(true);
}
public static void main(String[] args) {

Aman awt = new Aman();
}
}
```

OUTPUT:-

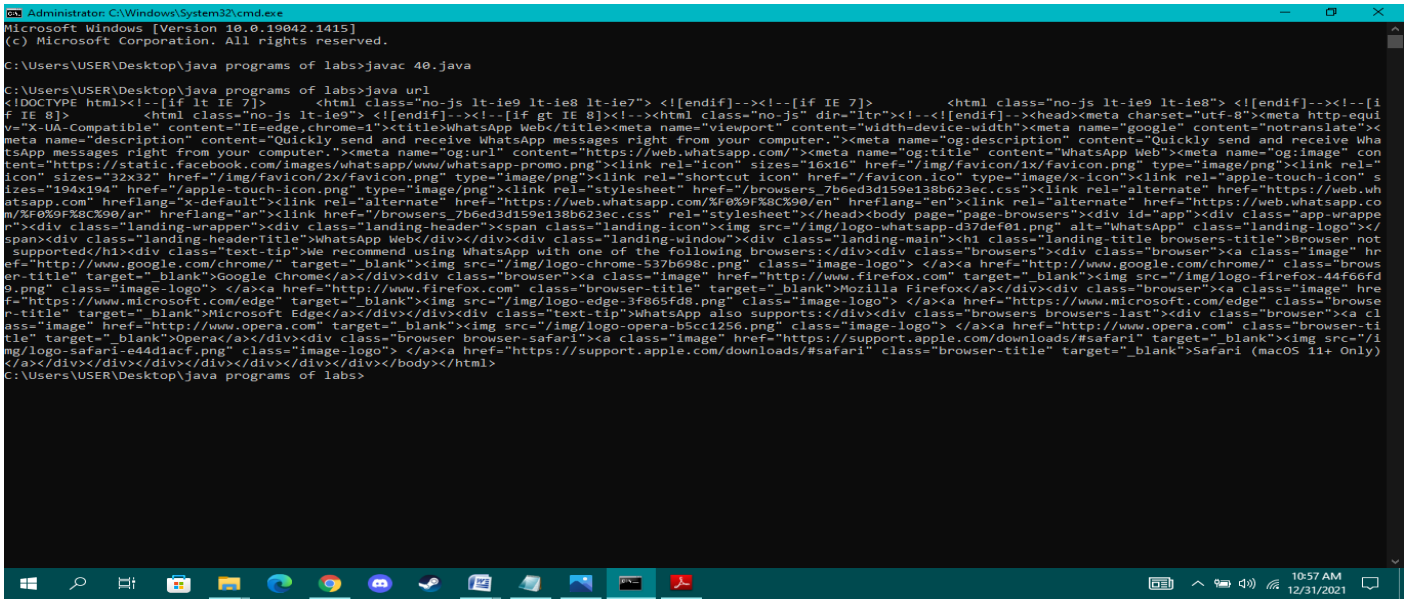


PROGRAM NO:-41

Demonstrate the use of url connection:-

```
import java.io.*;
import java.net.*;
 class url {
public static void main(String[] args){
try{
URL url=new URL("https://web.whatsapp.com/");
URLConnection urlcon=url.openConnection();
InputStream stream=urlcon.getInputStream();
int i;
while((i=stream.read())!=-1){
System.out.print((char)i);
}
}catch(Exception e){System.out.println(e);}
}
}
```

OUTPUT:-



PROGRAM NO:-42

Demonstrate client server model.

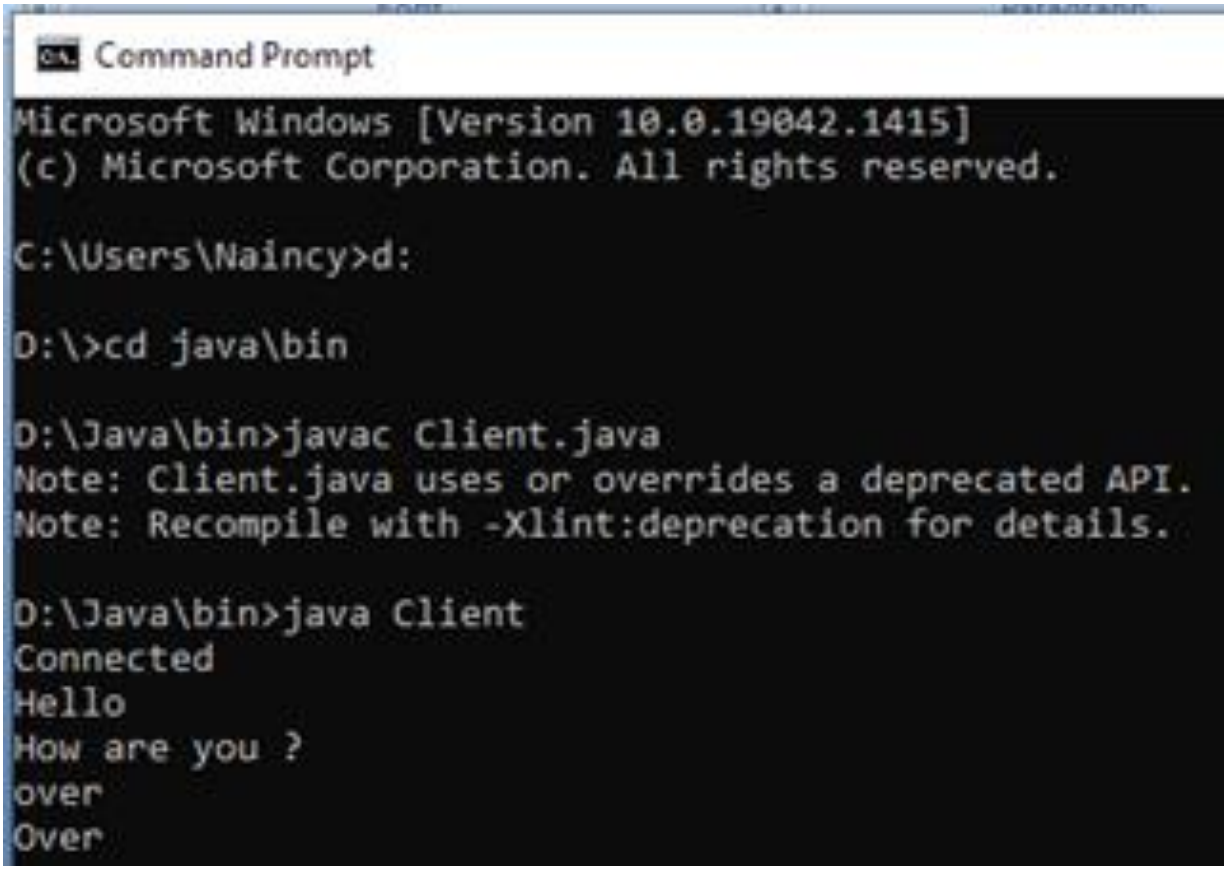
```
import java.net.*;
import java.io.*;

public class Client
{

   private Socket socket           = null;
   private DataInputStream  input   = null;
   private DataOutputStream out     = null;


   public Client(String address, int port)
   {

      try
      {
         socket = new Socket(address, port);
         System.out.println("Connected");


         input  = new DataInputStream(System.in);

         out    = new DataOutputStream(socket.getOutputStream());
      }
      catch(UnknownHostException u)
      {
         System.out.println(u);
      }
      catch(IOException i)
      {
         System.out.println(i);
      }


      String line = "";


      while (!line.equals("Over"))
      {
         try
         {
            line = input.readLine();
            out.writeUTF(line);
         }
         catch(IOException i)
```

```
         {
            System.out.println(i);
         }
      }


      try
      {
         input.close();
         out.close();
         socket.close();
      }
      catch(IOException i)
      {
         System.out.println(i);
      }
   }

   public static void main(String args[])
   {
      Client client = new Client("127.0.0.1", 5000);
   }
}
```

```
import java.net.*;
import java.io.*;

public class Server
{
   private Socket         socket   = null;
   private ServerSocket    server   = null;
   private DataInputStream in       =  null;


   public Server(int port)
   {

      try
      {
         server = new ServerSocket(port);
         System.out.println("Server started");

         System.out.println("Waiting for a client ...");

         socket = server.accept();
```

```
            System.out.println("Client accepted");

            in = new DataInputStream(
               new BufferedInputStream(socket.getInputStream()));

            String line = "";

            while (!line.equals("Over"))
            {
               try
               {
                  line = in.readUTF();
                  System.out.println(line);

               }
               catch(IOException i)
               {
                  System.out.println(i);
               }
            }
            System.out.println("Closing connection");

            socket.close();
            in.close();
         }
         catch(IOException i)
         {
            System.out.println(i);
         }
      }

      public static void main(String args[])
      {
         Server server = new Server(5000);
      }
   }
```

OUTPUT:-


Command Prompt
Microsoft Windows [Version 10.0.19042.1415]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Naincy>d:
D:\>cd java\bin
D:\Java\bin>javac Client.java
Note: Client.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
D:\Java\bin>java Client
Connected
Hello
How are you ?
over
Over

PROGRAM NO:-43

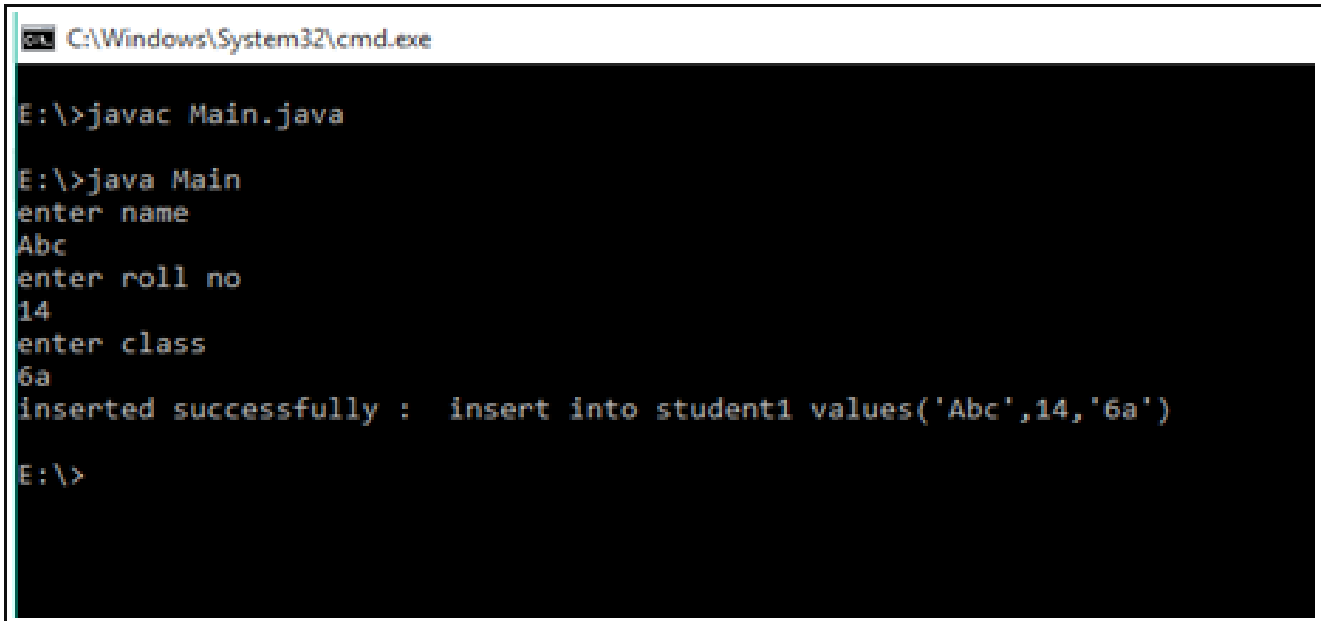
Demonstrate JDBC

```
importjava.sql.*;
// Importing required classes
importjava.util.*;

// Main class
class Main {

    // Main driver method
    public static void main(String a[])
    {

        // Creating the connection using Oracle DB
        // Note: url syntax is standard, so do grasp
        String url = "jdbc:oracle:thin:@localhost:1521:xe";

        // Usernamer and password to access DB
        // Custom initialization
        String user = "system";
        String pass = "12345";

        // Entering the data
        Scanner k = new Scanner(System.in);

        System.out.println("enter name");
        String name = k.next();

        System.out.println("enter roll no");
        int roll = k.nextInt();

        System.out.println("enter class");
        String cls = k.next();
```

```
// Inserting data using SQL query
String sql = "insert into student1 values('" + name
          + "'," + roll + ",'" + cls + "')";

// Connection class object
Connection con = null;

// Try block to check for exceptions
try {

   // Registering drivers
   DriverManager.registerDriver(
      new oracle.jdbc.OracleDriver());

   // Reference to connection interface
   con = DriverManager.getConnection(url, user,
                                     pass);

   // Creating a statement
   Statement st = con.createStatement();

   // Executing query
   int m = st.executeUpdate(sql);
   if (m == 1)
      System.out.println(
         "inserted successfully : " + sql);
   else
      System.out.println("insertion failed");

   // Closing the connections
   con.close();
}

// Catch block to handle exceptions
```

```
        catch (Exception ex) {
            // Display message when exceptions occurs
            System.err.println(ex);
        }
    }
}
```

OUTPUT:-



PROGRAM NO:-44

Demonstrate RMI:-

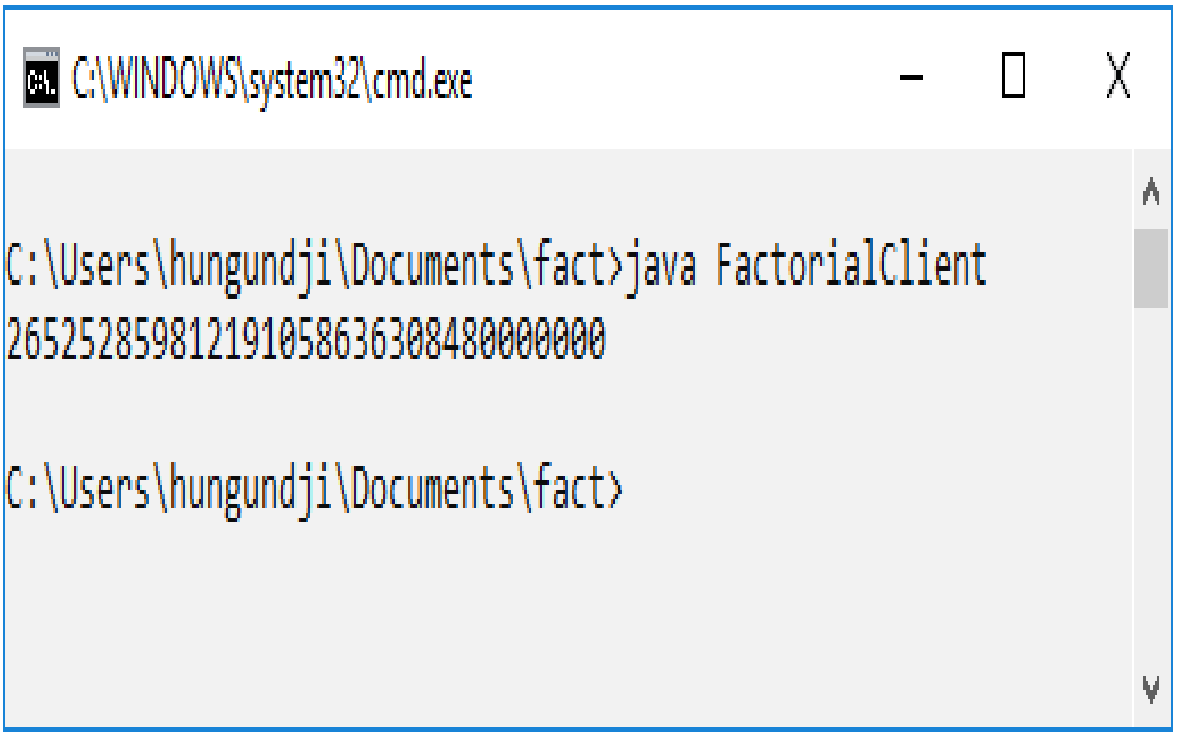
```
import java.math.BigInteger;

// Creating an Interface
public interface Factorial
   extends java.rmi.Remote {

   // Declaring the method
   public BigInteger fact(int num)
      throws java.rmi.RemoteException;
}
import java.math.BigInteger;

// Extends and Implement the class
// and interface respectively
public class FactorialImpl
   extends java.rmi.server.UnicastRemoteObject
   implements Factorial {

   // Constructor Declaration
   public FactorialImpl()
      throws java.rmi.RemoteException
   {
      super();
   }

   // Calculation for the problem statement
   // Implementing the method fact()
   // to find factorial of a number
```

```
    public BigInteger fact(int num)
        throws java.rmi.RemoteException
    {
        BigInteger factorial = BigInteger.ONE;

        for (int i = 1; i <= num; ++i) {
            factorial = factorial
                        .multiply(
                            BigInteger
                                .valueOf(i));
        }
        return factorial;
    }
}
import java.rmi.Naming;

public class FactorialServer {

    // Implement the constructor of the class
    public FactorialServer()
    {
        try {
            // Create a object reference for the interface
            Factorial c = new FactorialImpl();

            // Bind the localhost with the service
            Naming.rebind("rmi:// localhost/FactorialService", c);
        }
        catch (Exception e) {
            // If any error occur
            System.out.println("ERR: " + e);
```

```
      }
   }

   public static void main(String[] args)
   {
      // Create an object
      new FactorialServer();
   }
}

import java.net.MalformedURLException;
import java.rmi.Naming;
import java.rmi.NotBoundException;
import java.rmi.RemoteException;

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

      try {
         // Create an remote object with the same name
         // Cast the lookup result to the interface
         Factorial c =(Factorial);
         Naming.lookup("rmi:// localhost/FactorialService");

         // Call the method for the results
         System.out.println(c.fact(30));
      }

      // If any error occur
      catch (MalformedURLException murle) {
```

```
            System.out.println("\nMalformedURLException: " + murle);
        }

        catch (RemoteException re) {
            System.out.println("\nRemoteException: "+ re);
        }

        catch (NotBoundException nbe) {
            System.out.println("\nNotBoundException: "  + nbe);
        }

        catch (java.lang.ArithmeticException ae) {
            System.out.println("\nArithmeticException: " + ae);
        }
    }
```

OUTPUT:-



}