Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2022-23

(310919) Java Programming Laboratory

****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 1 **Date of Implementation:** 2. 5. 23

****

Q1. Write down the steps to implement small program from command prompt.

Ans.

Before running a Java program, we need to ensure that Java is installed in the system and the path is properly set. If the path is not properly set, we cannot run the Java program**.**

1. Open the notepad and write a Java program into it. Write a Java program that you want to compile and run.
2. Save the Java program by using the class name followed by .java extension.
3. To compile and run a Java program, open the Command Prompt by pressing Windows Key + R, type cmd and press enter key or click on the Ok button. It opens the Command Prompt window.
4. In the Command Prompt window, write the following commands.
   cd\
   cd <directory name where java file is stored>
5. To compile the Java program type the following command:
   <program name>.java
   eg: demo.java
6. When we compile a Java program without any error, it creates a .class file with the same name as the file name at the same location (where the program is saved).
7. To run the Java program, type the following command.
   <program name>
   eg: demo

Q2. Write down the steps to implement small program in Eclipse.

Ans.

1. Open Eclipse and click File > New > Java Project.
2. Provide the Project Name and click on the Finish button.
3. In the Package Explorer (left-hand side of the window) select the project which you have created.
4. Right-click on the src folder, select New > Class from the submenu. Provide the Class name and click on Finish button.
5. Write the program and save it.
6. Now, press Ctrl+F11 or click on the Run menu and select Run or click on Run button to get output.

****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 1 **Date of Implementation:** 9. 5. 23

****

Q3. Write a Java program to find Armstrong's number An Armstrong number is a 3 digit number for which the sum of cube of its digits is equal to the number itself. (e.g. 153).

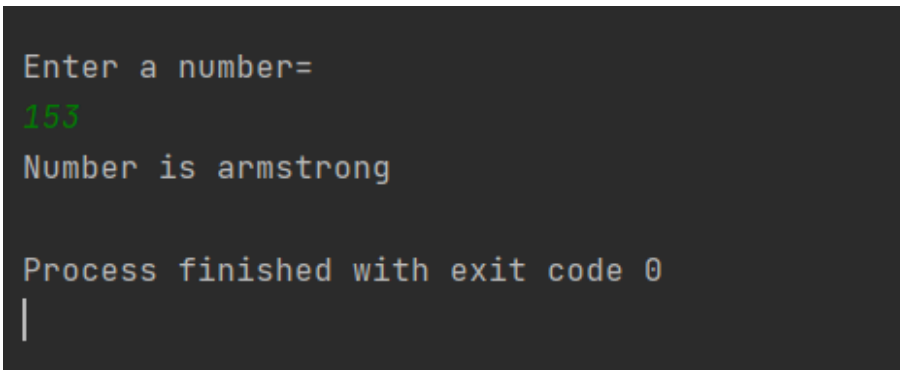
Program:

```
import java.util.Scanner;

public class armstrong {

   public static void main(String[] args){
      Scanner sc = new Scanner(System.in);
      int num, sum=0, r;
      System.out.println("\nEnter a number= ");
      num = sc.nextInt();
      int n=num;
      while(num>0){
         r=num%10;
         sum=sum+(r*r*r);
         num=num/10;
      }

      if (n==sum){
         System.out.println("Number is armstrong");
      }else{
         System.out.println("Number is not armstrong");
      }
   }
}
```

Output:

```
Enter a number=
153
Number is armstrong

Process finished with exit code 0
```

Screenshot:



****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 1 **Date of Implementation:** 9. 5. 23

****

Q4. Write a class of Account Holder, which has a Date object. The class should have following functionality:
• To show the date of account creation, balance, account number and name of account holder.
• To deposit and withdraw money.
• To display the current status and transaction.
• Display balance of both saving, current account using one method

Program:

```
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.Scanner;

class actions{

   String name;
   int ac_num;
   float s_balance, c_balance;
   Scanner sc = new Scanner(System.in);
   DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy/MM/dd HH:mm:ss");
   LocalDateTime now = LocalDateTime.now();

   public void get(){
      System.out.println("Enter account holder name: ");
      name = sc.nextLine();

      System.out.println("Enter account number: ");
      ac_num = sc.nextInt();

      System.out.println("Enter savings balance: ");
      s_balance = sc.nextFloat();

      System.out.println("Enter current balance: ");
      c_balance = sc.nextFloat();

   }

   public void display(){
      System.out.println("Account holder name: " + name);
      System.out.println("Account number: " + ac_num);
      System.out.println("Savings Balance: " + s_balance);
      System.out.println("Savings Balance: " + c_balance);
      System.out.println("Date of account creation: "+ dtf.format(now));
   }

   public void deposit(){
      float amt;
      int x;
```

```
        do{
            System.out.println("Enter 1 to deposit to in savings \nEnter 2 to deposit to in current \nEnter 3 to
exit");
            x = sc.nextInt();
            switch (x){
                case 1:
                    System.out.println("Enter amount to deposit: ");
                    amt = sc.nextFloat();

                    s_balance += amt;
                    System.out.println("New balance= " + s_balance);
                    break;

                case 2:
                    System.out.println("Enter amount to deposit: ");
                    amt = sc.nextFloat();

                    c_balance += amt;
                    System.out.println("New balance= " + c_balance);
                    break;

                case 3:
                    break;

                default:
                    System.out.println("Wrong choice");
                    break;

            }
        }while (x != 3);



    }

    public void withdraw(){

        float amt;
        int x;

       do{
            System.out.println("Enter 1 to withdraw from savings \nEnter 2 to withdraw from current
\nEnter 3 to exit ");
            x = sc.nextInt();
            switch (x){
                case 1:
                    System.out.println("Enter amount to withdraw: ");
                    amt = sc.nextFloat();

                    if(s_balance>=amt){
                        s_balance -= amt;
                        System.out.println("New balance= " + s_balance);
                    }else{
                        System.out.println("Balance is insufficient");
                    }
                    break;
```

```
            case 2:
               System.out.println("Enter amount to withdraw: ");
               amt = sc.nextFloat();

               if(c_balance>=amt){
                  c_balance -= amt;
                  System.out.println("New balance= " + c_balance);
               }else{
                  System.out.println("Balance is insufficient");
               }
               break;

            case 3:
               break;

            default:
               System.out.println("Wrong choice");
               break;

         }
      } while (x != 3);

   }

   public void show(){
      System.out.println("Savings A/c balance: " + s_balance);
      System.out.println("Current A/c balance: " + c_balance);

   }

}



public class account_holder {

   public static void main(String[] args){
      int n;
      Scanner sc = new Scanner(System.in);
      actions a = new actions();
      a.get();

      do{
         System.out.println("\nEnter 1 to display details");
         System.out.println("Enter 2 to deposit");
         System.out.println("Enter 3 to withdraw");
         System.out.println("Enter 4 to see savings and current balance");
         System.out.println("Enter 5 to exit");
         n = sc.nextInt();

         switch (n){
            case 1:
               a.display();
               break;

            case 2:
```

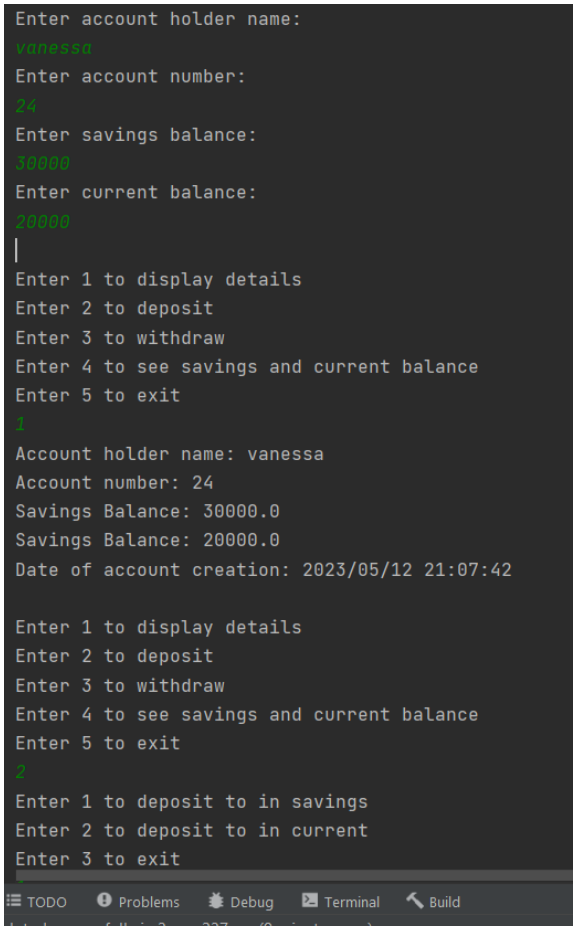
```
                a.deposit();
                break;

            case 3:
                a.withdraw();
                break;

            case 4:
                a.show();
                break;

            case 5:
                break;

            default:
                System.out.println("Wrong choice");

        }

    }while(n!=5);
  }
}
```

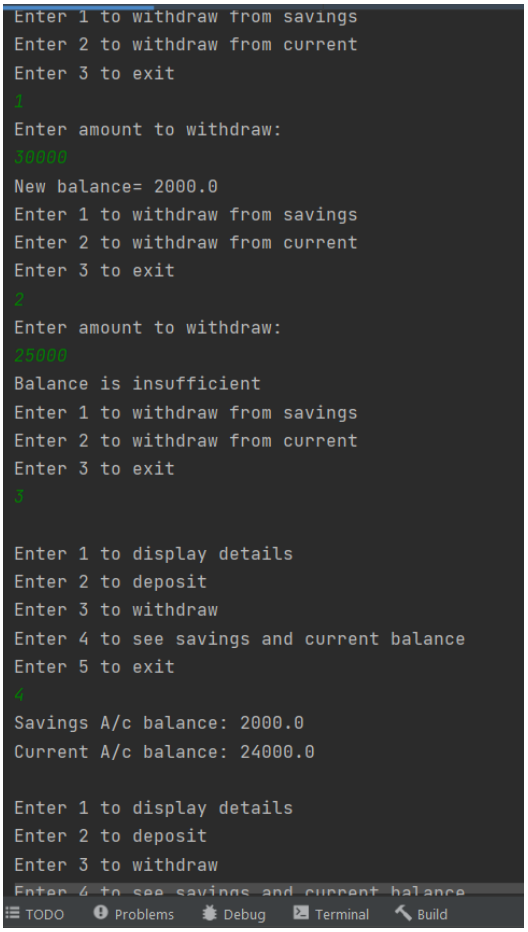
Output:

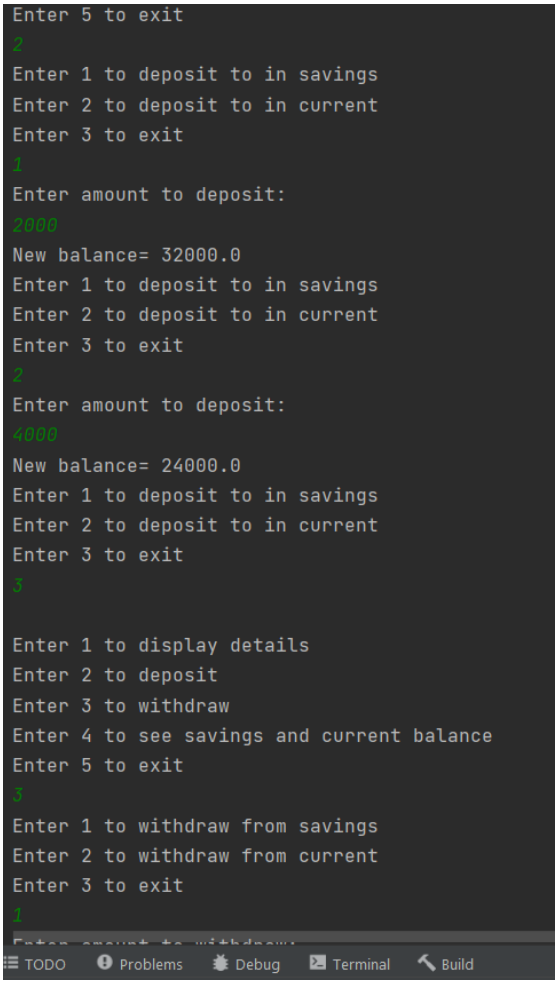
```
Enter account holder name:
vanessa
Enter account number:
24
Enter savings balance:
30000
Enter current balance:
20000

Enter 1 to display details
Enter 2 to deposit
Enter 3 to withdraw
Enter 4 to see savings and current balance
Enter 5 to exit
1
Account holder name: vanessa
Account number: 24
Savings Balance: 30000.0
Savings Balance: 20000.0
Date of account creation: 2023/05/12 21:07:42

Enter 1 to display details
Enter 2 to deposit
Enter 3 to withdraw
Enter 4 to see savings and current balance
Enter 5 to exit
2
Enter 1 to deposit to in savings
Enter 2 to deposit to in current
Enter 3 to exit
1
```

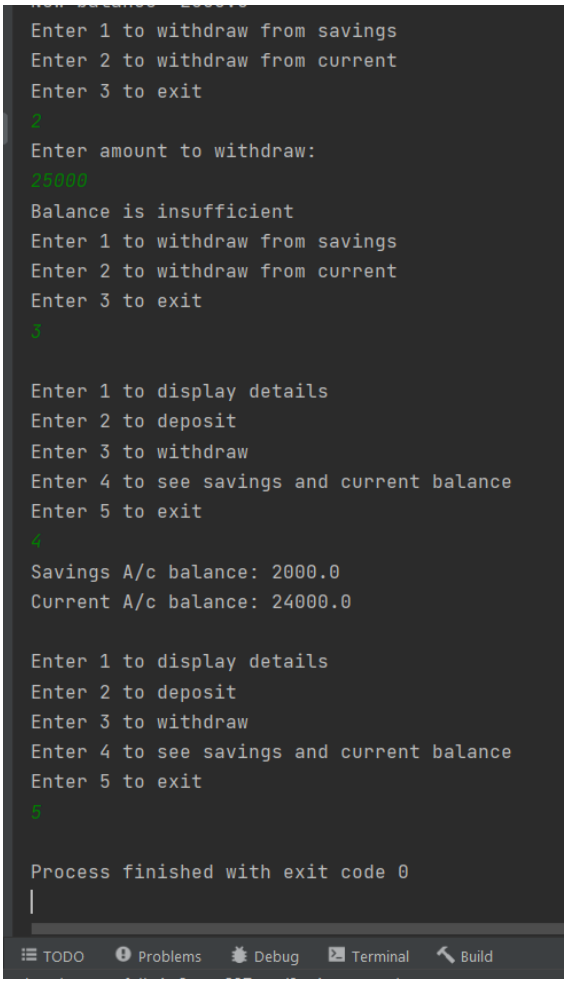
```
Enter amount to deposit:
2000
New balance= 32000.0
Enter 1 to deposit to in savings
Enter 2 to deposit to in current
Enter 3 to exit
2
Enter amount to deposit:
4000
New balance= 24000.0
Enter 1 to deposit to in savings
Enter 2 to deposit to in current
Enter 3 to exit
3

Enter 1 to display details
Enter 2 to deposit
Enter 3 to withdraw
Enter 4 to see savings and current balance
Enter 5 to exit
3
Enter 1 to withdraw from savings
Enter 2 to withdraw from current
Enter 3 to exit
1
Enter amount to withdraw:
30000
New balance= 2000.0
Enter 1 to withdraw from savings
Enter 2 to withdraw from current
Enter 3 to exit
2
Enter amount to withdraw:
25000
Balance is insufficient
Enter 1 to withdraw from savings
Enter 2 to withdraw from current
Enter 3 to exit
3

Enter 1 to display details
Enter 2 to deposit
Enter 3 to withdraw
Enter 4 to see savings and current balance
Enter 5 to exit
4
Savings A/c balance: 2000.0
Current A/c balance: 24000.0

Enter 1 to display details
Enter 2 to deposit
Enter 3 to withdraw
Enter 4 to see savings and current balance
Enter 5 to exit
5

Process finished with exit code 0
```

Screenshot:


Enter account holder name:
vanessa
Enter account number:
24
Enter savings balance:
30000
Enter current balance:
20000
Enter 1 to display details
Enter 2 to deposit
Enter 3 to withdraw
Enter 4 to see savings and current balance
Enter 5 to exit
1
Account holder name: vanessa
Account number: 24
Savings Balance: 30000.0
Savings Balance: 20000.0
Date of account creation: 2023/05/12 21:07:42
Enter 1 to display details
Enter 2 to deposit
Enter 3 to withdraw
Enter 4 to see savings and current balance
Enter 5 to exit
2
Enter 1 to deposit to in savings
Enter 2 to deposit to in current
Enter 3 to exit
TODO
Problems
Debug
Terminal
Build


Enter 1 to withdraw from savings
Enter 2 to withdraw from current
Enter 3 to exit
1
Enter amount to withdraw:
30000
New balance= 2000.0
Enter 1 to withdraw from savings
Enter 2 to withdraw from current
Enter 3 to exit
2
Enter amount to withdraw:
25000
Balance is insufficient
Enter 1 to withdraw from savings
Enter 2 to withdraw from current
Enter 3 to exit
3
Enter 1 to display details
Enter 2 to deposit
Enter 3 to withdraw
Enter 4 to see savings and current balance
Enter 5 to exit
4
Savings A/c balance: 2000.0
Current A/c balance: 24000.0
Enter 1 to display details
Enter 2 to deposit
Enter 3 to withdraw
TODO
Problems
Debug
Terminal
Build


Enter 5 to exit
2
Enter 1 to deposit to in savings
Enter 2 to deposit to in current
Enter 3 to exit
1
Enter amount to deposit:
2000
New balance= 32000.0
Enter 1 to deposit to in savings
Enter 2 to deposit to in current
Enter 3 to exit
2
Enter amount to deposit:
4000
New balance= 24000.0
Enter 1 to deposit to in savings
Enter 2 to deposit to in current
Enter 3 to exit
3
Enter 1 to display details
Enter 2 to deposit
Enter 3 to withdraw
Enter 4 to see savings and current balance
Enter 5 to exit
3
Enter 1 to withdraw from savings
Enter 2 to withdraw from current
Enter 3 to exit
1
TODO
Problems
Debug
Terminal
Build


Enter 1 to withdraw from savings
Enter 2 to withdraw from current
Enter 3 to exit
2
Enter amount to withdraw:
25000
Balance is insufficient
Enter 1 to withdraw from savings
Enter 2 to withdraw from current
Enter 3 to exit
3
Enter 1 to display details
Enter 2 to deposit
Enter 3 to withdraw
Enter 4 to see savings and current balance
Enter 5 to exit
4
Savings A/c balance: 2000.0
Current A/c balance: 24000.0
Enter 1 to display details
Enter 2 to deposit
Enter 3 to withdraw
Enter 4 to see savings and current balance
Enter 5 to exit
5
Process finished with exit code 0
TODO
Problems
Debug
Terminal
Build

****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 1 **Date of Implementation:** 9. 5. 23

****

Q5. Create a class circle and use inheritance to create another class cylinder from it. Create a method for area and volume.

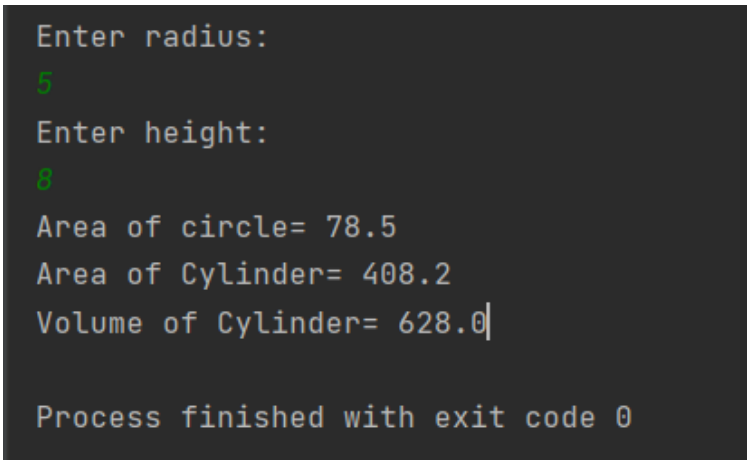
Program:

```
import java.util.Scanner;

class Circle{

   public void circle_area(float r){
      float area;

      area = (float) (3.14 * r * r);
      System.out.println("Area of circle= " + area);

   }
}

public class Cylinder extends Circle{

   public static void main(String[] args){
      Scanner sc = new Scanner(System.in);
      Cylinder c = new Cylinder();
      float radius, height, area_cyl, vol;

      System.out.println("Enter radius: ");
      radius=sc.nextFloat();
      System.out.println("Enter height: ");
      height=sc.nextFloat();

      c.circle_area(radius);

      area_cyl = (float) ((2 * 3.14 * radius * radius ) + (2 * 3.14 * radius *height));
      vol = (float) (3.14 * radius * radius * height);

      System.out.println("Area of Cylinder= " + area_cyl);
      System.out.println("Volume of Cylinder= " + vol);

   }
}
```

Output:

```
Enter radius:
5
Enter height:
8
Area of circle= 78.5
Area of Cylinder= 408.2
Volume of Cylinder= 628.0

Process finished with exit code 0
```

Screenshot:



****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 1 **Date of Implementation:** 12. 5. 23

****

Q6. Create a class named "Member" having the following members:
Data members
1 - Name
2 - Age
3 - Phone number
4 - Address
5 - Salary
It also has a method named "printSalary" which prints the salary of the members.
Two classes "Employee" and "Manager" inherits the "Member" class. The "Employee" and "Manager" classes have data members "specialization" and "department" respectively. Now, assign name, age, phone number, address and
salary to an employee and a manager by making an object of both of these classes and print the same.

Program:

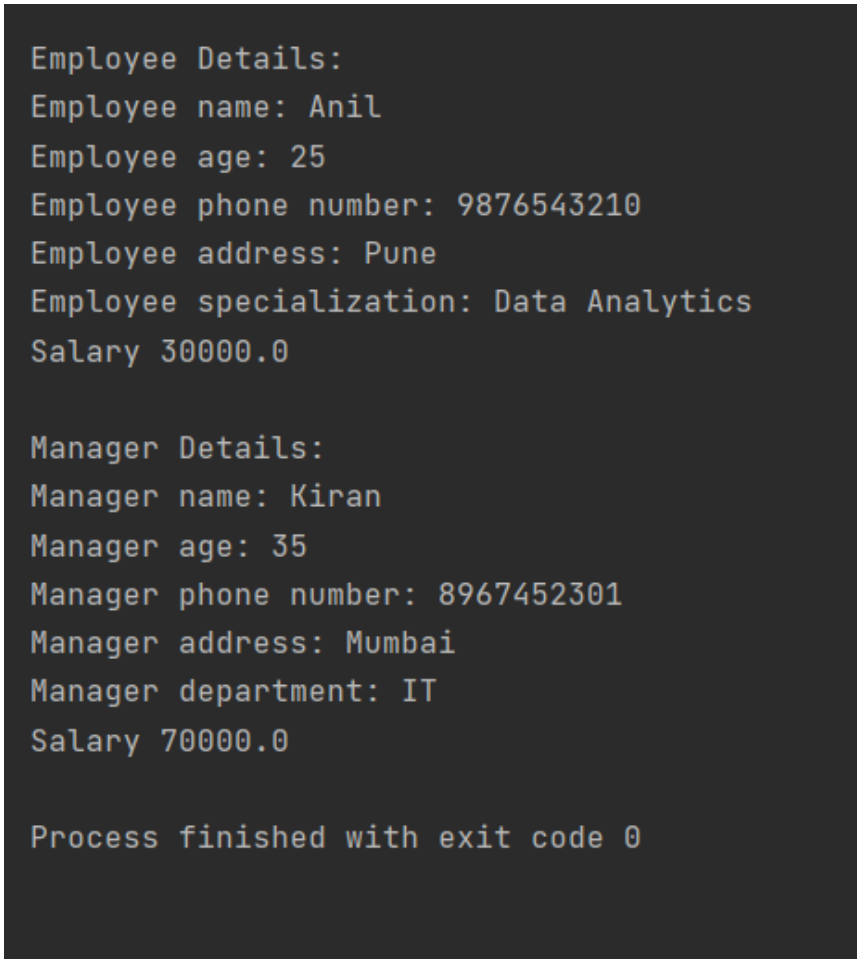
```
class Member {
    private String name;
    private int age;
    private String phoneNumber;
    private String address;
    private double salary;

    public Member(String name, int age, String phoneNumber, String address, double salary) {
        this.name = name;
        this.age = age;
        this.phoneNumber = phoneNumber;
        this.address = address;
        this.salary = salary;
    }

    public void printSalary() {
        System.out.println("Salary " + salary);
    }
}

class Employee extends Member {

    public Employee(String name, int age, String phoneNumber,
                    String address, double salary, String specialization) {
        super(name, age, phoneNumber, address, salary);

        System.out.println("Employee name: " + name );
        System.out.println("Employee age: " + age);
        System.out.println("Employee phone number: " + phoneNumber);
        System.out.println("Employee address: " + address);
```

```
        System.out.println("Employee specialization: " + specialization);
    }

}

class Manager extends Member{

    public Manager(String name, int age, String phoneNumber,
                   String address, double salary, String department) {
        super(name, age, phoneNumber, address, salary);
        System.out.println("Manager name: " + name );
        System.out.println("Manager age: " + age);
        System.out.println("Manager phone number: " + phoneNumber);
        System.out.println("Manager address: " + address);
        System.out.println("Manager department: " + department);
    }
}

public class company {
    public static void main(String[] args) {
        System.out.println("\nEmployee Details: ");
           Employee employee = new Employee("Anil", 25, "9876543210", "Pune", 30000.00, "Data
Analytics");
        employee.printSalary();

        System.out.println("\nManager Details: ");
        Manager manager = new Manager("Kiran", 35, "8967452301", "Mumbai", 70000.00, "IT");
        manager.printSalary();
    }
}
```

Output:

```
Employee Details:
Employee name: Anil
Employee age: 25
Employee phone number: 9876543210
Employee address: Pune
Employee specialization: Data Analytics
Salary 30000.0

Manager Details:
Manager name: Kiran
Manager age: 35
Manager phone number: 8967452301
Manager address: Mumbai
Manager department: IT
Salary 70000.0

Process finished with exit code 0
```

Screenshot:


Employee Details:
Employee name: Anil
Employee age: 25
Employee phone number: 9876543210
Employee address: Pune
Employee specialization: Data Analytics
Salary 30000.0
Manager Details:
Manager name: Kiran
Manager age: 35
Manager phone number: 8967452301
Manager address: Mumbai
Manager department: IT
Salary 70000.0
Process finished with exit code 0

****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 1 **Date of Implementation:** 12. 5. 23

****

Q7. Write a Java program to calculate salary using packages. Creates a package employee and create a class Emp. Data members are name, employee id, category, basic pay, HRA, DA, net pay, provident fund, gross pay, income tax, and
allowance. Calculate the values in methods. Call the methods to perform and print values

Program:

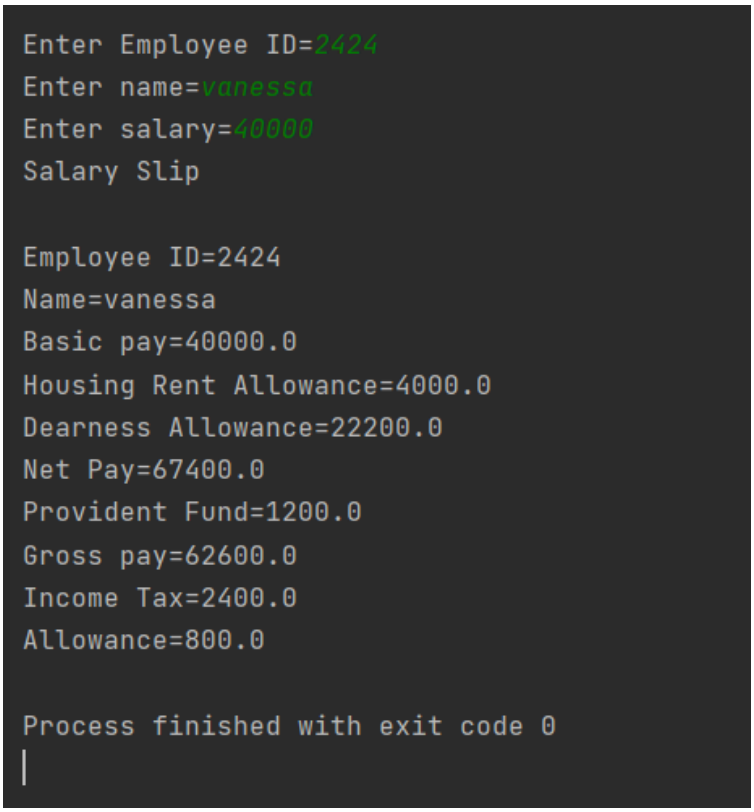
```
package employee;

import java.util.Scanner;

public class Emp {

   int emp_id;
   String name;
   float salary, hra, da, np, pf, gp, it, al;
   Emp()
   {
      emp_id=0;
      name = "";
      salary = 0.0f;
      hra = 0.0f;
      da = 0.0f;
      np = 0.0f;
      gp = 0.0f;
      it = 0.0f;
      al = 0.0f;
   }
   void get()
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("\nEnter Employee ID=");
      emp_id = sc.nextInt();
      System.out.print("Enter name=");
      name = sc.next();
      System.out.print("Enter salary=");
      salary = sc.nextFloat();
   }
   void calculate()
   {
      hra = salary * 0.1f;
      da = salary * 0.555f;
      pf = salary * 0.03f;
      it = salary * 0.06f;
      al = salary * 0.02f;
      gp = salary + hra + da - pf - it ;
      np = salary + hra + da + pf;
```

```
    }
    void display()
    {
        System.out.println("Salary Slip\n");
        System.out.println("Employee ID="+emp_id);
        System.out.println("Name="+name);
        System.out.println("Basic pay="+salary);
        System.out.println("Housing Rent Allowance="+hra);
        System.out.println("Dearness Allowance="+da);
        System.out.println("Net Pay="+np);
        System.out.println("Provident Fund="+pf);
        System.out.println("Gross pay="+gp);
        System.out.println("Income Tax="+it);
        System.out.println("Allowance="+al);
    }
    public static void main(String args[])
    {
        Emp e = new Emp();
        e.get();
        e.calculate();
        e.display();
    }
}
```

Output:

```
Enter Employee ID=2424
Enter name=vanessa
Enter salary=40000
Salary Slip

Employee ID=2424
Name=vanessa
Basic pay=40000.0
Housing Rent Allowance=4000.0
Dearness Allowance=22200.0
Net Pay=67400.0
Provident Fund=1200.0
Gross pay=62600.0
Income Tax=2400.0
Allowance=800.0

Process finished with exit code 0
```

Screenshot:


Enter Employee ID=2424
Enter name=vanessa
Enter salary=40000
Salary Slip
Employee ID=2424
Name=vanessa
Basic pay=40000.0
Housing Rent Allowance=4000.0
Dearness Allowance=22200.0
Net Pay=67400.0
Provident Fund=1200.0
Gross pay=62600.0
Income Tax=2400.0
Allowance=800.0
Process finished with exit code 0

****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 1 **Date of Implementation:** 12. 5. 23

****

Q8. Write program to implementing the Operations of stack and queue using interface.

Program:

```
import java.util.Scanner;

interface stack{
   void push(int item);
   int pop();
   void display();
}

interface queue{
   void enqueue(int item);
   void dequeue();
   void show();
}

class StackImplementation implements stack{
   private int stk[ ];
   private int top;
   StackImplementation(int size)
   {
      stk=new int[size];
      top=-1;
   }
   public void push(int item)
   {
      if(top>=stk.length-1)
         System.out.println("Stack Overflow");
      else {
         top++;
         stk[top]=item;
      }

   }
   public int pop()
   {
      if(top<0)
      {
         System.out.println("Stack Underflows");
         return 0;
      }
      else
         return stk[top--];
   }
```

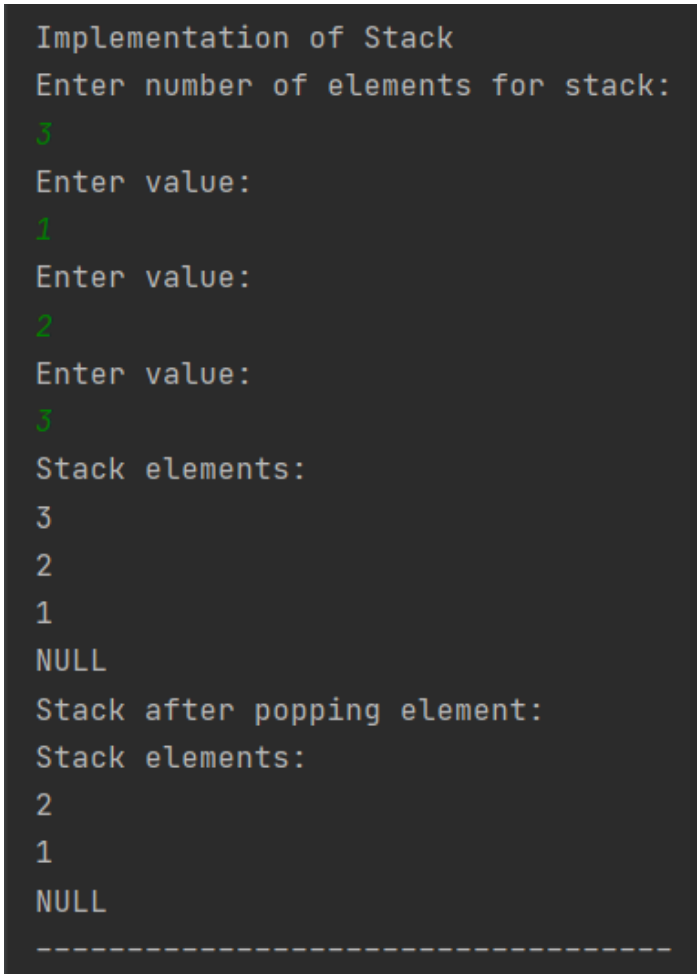
```
    public void display(){
        if(top>=0) {
            for(int i=top; i>=0; i--){
                System.out.println(stk[i]);
            }
            System.out.println("NULL");
        } else
            System.out.println("Stack is empty");

    }
}

class QueueImplementation implements queue{

    private static int front, rear, qsize;
    private static int[] queue;

    QueueImplementation(int size) {
        front = rear = 0;
        qsize = size;
        queue = new int[qsize];
    }

    public void enqueue(int item) {
        if (qsize == rear) {
            System.out.println("\nQueue is full\n");
        }

        else {
            queue[rear] = item;
            rear++;
        }
    }

    public void dequeue() {
        if (front == rear) {
            System.out.println("\nQueue is empty\n");
        }
        else {
            for (int i = 0; i < rear - 1; i++) {
                queue[i] = queue[i + 1];
            }
            if (rear < qsize)
                queue[rear] = 0;

          rear--;
        }
    }

    public void show() {
        int i;
        if (front == rear) {
            System.out.println("Queue is Empty\n");
            return;
        }

        for (i = front; i < rear; i++) {
```

```
            System.out.print(queue[i] + " ");
        }
    }
}

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

        System.out.println("\nImplementation of Stack");
        System.out.println("Enter number of elements for stack: ");
        n = sc.nextInt();
        StackImplementation s =new StackImplementation(n);

        for(int i=0; i<n; i++){
            System.out.println("Enter value: ");
            value =  sc.nextInt();
            s.push(value);
        }

        System.out.println("Stack elements: ");
        s.display();

        System.out.println("Stack after popping element: ");
        s.pop();

        System.out.println("Stack elements: ");
        s.display();

        System.out.println("-----------------------------------");

        System.out.println("Implementation of Queue");
        System.out.println("Enter number of elements for queue: ");
        n = sc.nextInt();
        QueueImplementation q =new QueueImplementation(n);

        for(int i=0; i<n; i++){
            System.out.println("Enter value: ");
            value =  sc.nextInt();
            q.enqueue(value);
        }

        System.out.println("Queue elements: ");
        q.show();

        System.out.println("\nQueue after dequeing element: ");
        q.dequeue();

        System.out.println("Queue elements: ");
        q.show();

        System.out.println("\n-----------------------------------");

    }
}
```

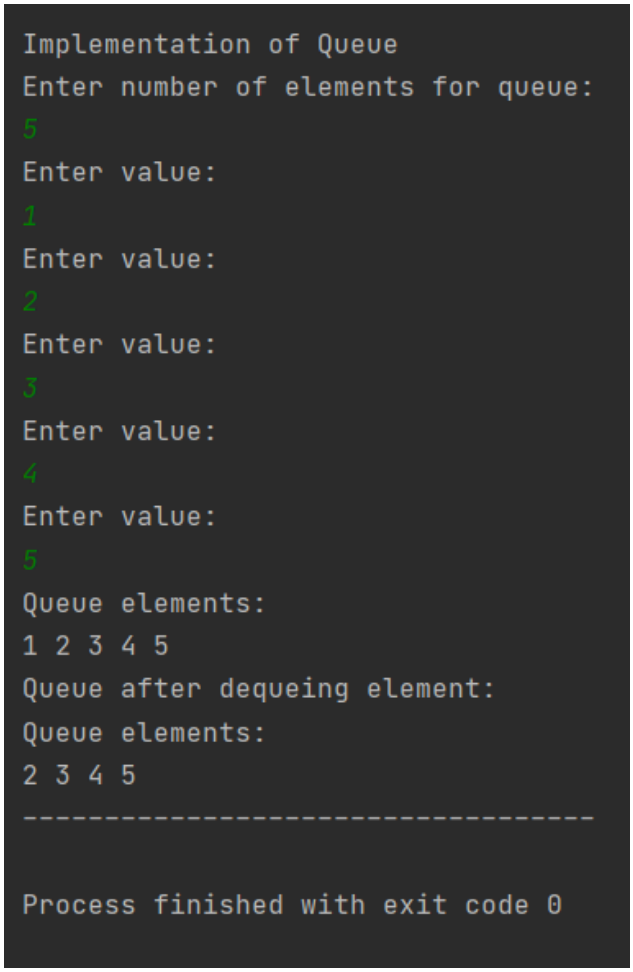
Output:

```
Implementation of Stack
Enter number of elements for stack:
3
Enter value:
1
Enter value:
2
Enter value:
3
Stack elements:
3
2
1
NULL
Stack after popping element:
Stack elements:
2
1
NULL
-----------------------------------
Implementation of Queue
Enter number of elements for queue:
5
Enter value:
1
Enter value:
2
Enter value:
3
Enter value:
4
Enter value:
5
Queue elements:
1 2 3 4 5
Queue after dequeing element:
Queue elements:
2 3 4 5
-----------------------------------

Process finished with exit code 0
```

Screenshot:



```
Implementation of Stack
Enter number of elements for stack:
3
Enter value:
1
Enter value:
2
Enter value:
3
Stack elements:
3
2
1
NULL
Stack after popping element:
Stack elements:
2
1
NULL
----------------------------------
```



```
Implementation of Queue
Enter number of elements for queue:
5
Enter value:
1
Enter value:
2
Enter value:
3
Enter value:
4
Enter value:
5
Queue elements:
1 2 3 4 5
Queue after dequeing element:
Queue elements:
2 3 4 5
----------------------------------

Process finished with exit code 0
```

****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 1 **Date of Implementation:** 12. 5. 23

****

Q9. Write program to accept roll no. Marks from user. Throw user defined exception Marks out of Bound if marks are <0 or marks>100.

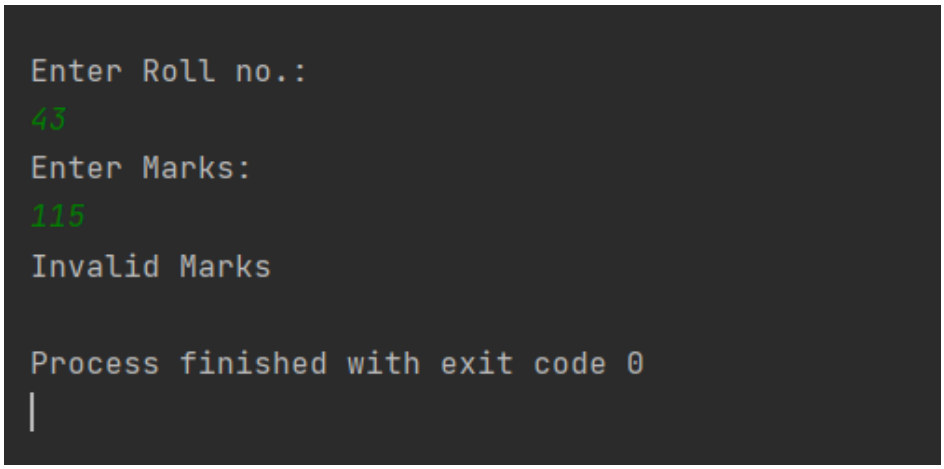
Program:

```
import java.util.Scanner;

class MarksOutOfBounds extends Exception{
   public void error()
   {
      System.out.println("Invalid Marks");
   }
}

class User_def_exception{

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

      Scanner sc = new Scanner(System.in);

      int m=0, roll_no;
      try{

         System.out.println("\nEnter Roll no.:");
         roll_no = sc.nextInt();

         System.out.println("Enter Marks:");
         m = sc.nextInt();


         if(m>100 || m<0)
            throw new MarksOutOfBounds();

         System.out.println("Roll no:"+roll_no);
         System.out.println("Marks:"+m);

      }

      catch(MarksOutOfBounds e){
         e.error();
      }
   }
}
```

Output:

```
Enter Roll no.:
43
Enter Marks:
115
Invalid Marks

Process finished with exit code 0
```

Screenshot:



****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 1 **Date of Implementation:** 12. 5. 23

****

Q10. Create an array of a list of employee. The list may contain a Manager of Sales Manager object. The program should display list of the salary of respective kind of employee in the array.

Program:

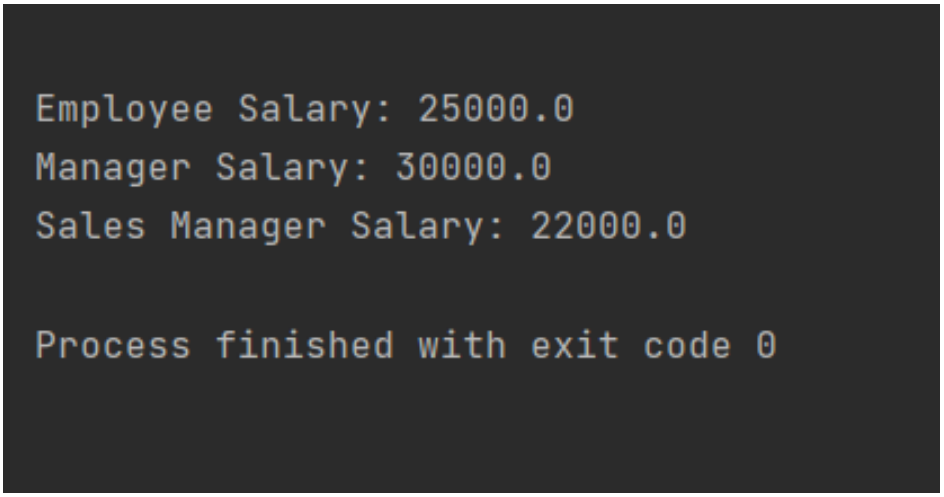
```
import java.util.ArrayList;
import java.util.List;

class emp {
   protected String name;
   protected double salary;

   public emp(String name, double salary) {
      this.name = name;
      this.salary = salary;
   }

   public double getSalary() {
      return salary;
   }
}

class Mng extends emp {
   private String department;

   public Mng(String name, double salary, String department) {
      super(name, salary);
      this.department = department;
   }

   public String getDepartment() {
      return department;
   }
}

class SalesManager extends Mng {
   private double bonus;

   public SalesManager(String name, double salary, String department, double bonus) {
      super(name, salary, department);
      this.bonus = bonus;
   }

   public double getBonus() {
      return bonus;
   }
```

```
    @Override
    public double getSalary() {
        return super.getSalary() + bonus;
    }
}

public class sales {
    public static void main(String[] args) {
        List<emp> employeeList = new ArrayList<>();
        employeeList.add(new emp("Bhushan", 25000));
        employeeList.add(new Mng("Raj", 30000, "IT"));
        employeeList.add(new SalesManager("Riya", 20000, "Sales", 2000));

        System.out.println("\n");
        for (emp employee : employeeList) {
            if (employee instanceof SalesManager) {
                System.out.println("Sales Manager Salary: " + employee.getSalary());
            } else if (employee instanceof Mng) {
                System.out.println("Manager Salary: " + employee.getSalary());
            } else {
                System.out.println("Employee Salary: " + employee.getSalary());
            }
        }
    }
}
```

Output:

```
Employee Salary: 25000.0
Manager Salary: 30000.0
Sales Manager Salary: 22000.0

Process finished with exit code 0
```

Screenshot:



****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 2 **Date of Implementation:** 16. 5. 23

****

Q1. Write a program to print “Hello Friends” and “Welcome to Java Programming” continuously on the screen in Java using threads. Add delay to its execution for 200ms.Also set priority to thread and display it.

Program:

```
class Hello extends Thread {
   synchronized public void run() {
      try {
         int i = 0;
         while (i < 5) {
            sleep(200);
            System.out.println("Hello!");
            i++;
         }
      } catch (Exception e) {
      }
   }
}

class Welcome extends Thread{
   synchronized public void run(){
      try{
         int i=0;
         while(i<5){
            sleep(200);
            System.out.println("Welcome to Java Programming!");
            i++;
         }
      }catch (Exception e){
      }
   }
}

public class threadImplementation {
   public static void main(String [] args){
      Hello t1 = new Hello();
      Welcome t2 = new Welcome();
      t1.setPriority(2);
      t2.setPriority(2);
      t1.start();
      t2.start();
   }
}
```

Output:

```
Hello!
Welcome to Java Programming!
Welcome to Java Programming!
Hello!
Hello!
Welcome to Java Programming!
Hello!
Welcome to Java Programming!
Welcome to Java Programming!
Hello!

Process finished with exit code 0
```

Screenshot:

****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 2 **Date of Implementation:** 16. 5. 23

****

Q2. Write a program to print even and odd numbers using two threads. Print the numbers in order, one thread only prints the even numbers and the other thread only prints the odd numbers.

Program:

```
import java.util.Scanner;
class OddThread extends Thread{
   int num;
   show display;

   public OddThread(int num, show display){
      this.num = num;
      this.display = display;
   }

  public void run(){
      int oddNumber = 1;

      while (oddNumber <= num){
         display.printOdd(oddNumber);
         oddNumber = oddNumber + 2;
      }
   }
}


class EvenThread extends Thread{
   int limit;
   show printer;

   public EvenThread(int limit, show printer){
      this.limit = limit;
      this.printer = printer;
   }

   @Override
   public void run(){
      int evenNumber = 2;
      while (evenNumber <= limit){
         printer.printEven(evenNumber);
         evenNumber = evenNumber + 2;
      }
   }
}

class show {
```

```
    boolean isOddPrinted = false;
    synchronized void printOdd(int number){
        while (isOddPrinted){
            try{
                wait();
            }
            catch (InterruptedException e){
                e.printStackTrace();
            }
        }

        System.out.println(Thread.currentThread().getName()+" : "+number);
        isOddPrinted = true;

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

        notify();
    }

     synchronized void printEven(int number)    {
        while (! isOddPrinted)        {
            try           {
                wait();
            }
            catch (InterruptedException e)           {
                e.printStackTrace();
            }
        }

        System.out.println(Thread.currentThread().getName()+" : "+number);

        isOddPrinted = false;

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


public class oddEvenThread
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number: ");
        int n = sc.nextInt();
```

```
        show printer = new show();

        OddThread oddThread = new OddThread(n, printer);
        oddThread.setName("Odd: ");
        EvenThread evenThread = new EvenThread(n, printer);
        evenThread.setName("Even: ");
        oddThread.start();
        evenThread.start();
    }
}
```

Output:

```
Enter number:
8
Odd:  : 1
Even:  : 2
Odd:  : 3
Even:  : 4
Odd:  : 5
Even:  : 6
Odd:  : 7
Even:  : 8

Process finished with exit code 0
```

Screenshot:

****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 2 **Date of Implementation:** 19. 5. 23

****

Q3. Create class Account which has attributes balance, account number with behavior display Balance, Withdraw and deposit Write a program to demonstration how the transactions in a bank can be carried out concurrent.

Program:

```
class Bank {
   static int total = 10000;


   static synchronized void withdrawn(String name, int withdrawal)
   {
      if (total >= withdrawal) {
         System.out.println(name + " withdrawn " + withdrawal);
         total = total - withdrawal;
         System.out.println("Balance after withdrawal: " + total);

         try {
            Thread.sleep(2000);
         }
         catch (InterruptedException e) {
            e.printStackTrace();
         }
      }

      else {
         System.out.println(name + " you can not withdraw " + withdrawal);
         System.out.println("your balance is: " + total);

         try {
            Thread.sleep(1000);
         }

         catch (InterruptedException e) {
            e.printStackTrace();
         }
      }
   }

   static synchronized void deposit(String name, int deposit)
   {
      System.out.println(name + " deposited " + deposit);
      total = total + deposit;
```

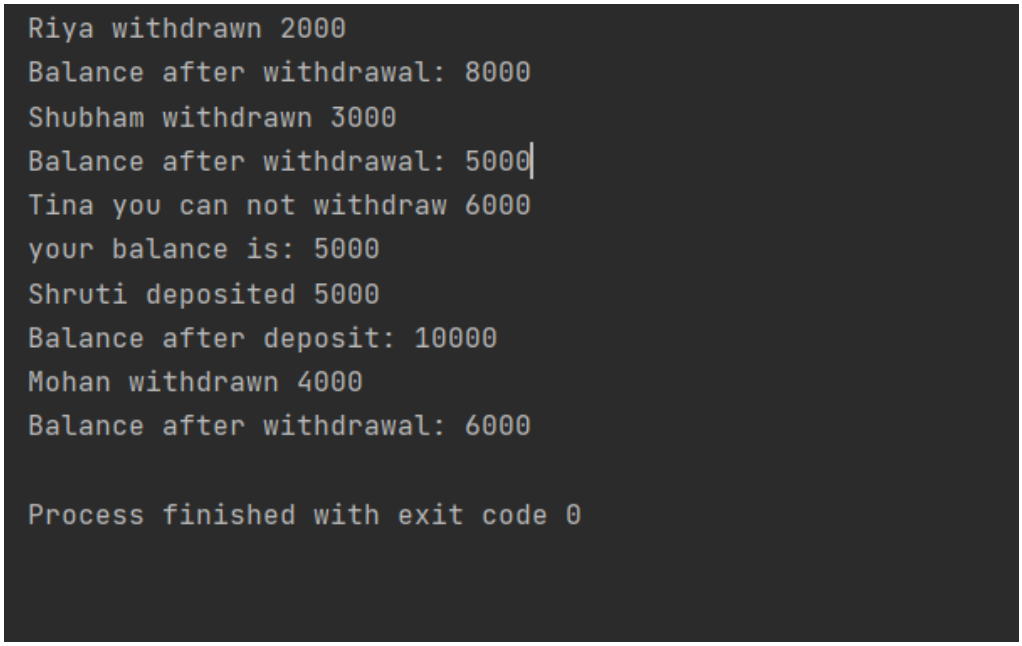
```
        System.out.println("Balance after deposit: " + total);

        try {
            Thread.sleep(1000);
        }

        catch (InterruptedException e) {

            e.printStackTrace();
        }
    }
}

class ThreadWithdrawal extends Thread {

    Bank object;
    String name;
    int dollar;

    // Constructor of this class
    ThreadWithdrawal(Bank ob, String name, int money)
    {
        this.object = ob;
        this.name = name;
        this.dollar = money;
    }


    public void run() { object.withdrawn(name, dollar); }
}


class ThreadDeposit extends Thread {

    Bank object;
    String name;
    int dollar;

    ThreadDeposit(Bank ob, String name, int money)
    {
        this.object = ob;
        this.name = name;
        this.dollar = money;
    }

    public void run() { object.deposit(name, dollar); }
}

class Account {


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

```
        Bank obj = new Bank();

        ThreadWithdrawal t1
                = new ThreadWithdrawal(obj, "Riya", 2000);
        ThreadWithdrawal t2
                = new ThreadWithdrawal(obj, "Mohan", 4000);
        ThreadDeposit t3
                = new ThreadDeposit(obj, "Shruti", 5000);
        ThreadWithdrawal t4
                = new ThreadWithdrawal(obj, "Tina", 6000);
        ThreadWithdrawal t5
                = new ThreadWithdrawal(obj, "Shubham", 3000);

        t1.start();
        t2.start();
        t3.start();
        t4.start();
        t5.start();
    }
}
```

Output:

```
Riya withdrawn 2000
Balance after withdrawal: 8000
Shubham withdrawn 3000
Balance after withdrawal: 5000
Tina you can not withdraw 6000
your balance is: 5000
Shruti deposited 5000
Balance after deposit: 10000
Mohan withdrawn 4000
Balance after withdrawal: 6000

Process finished with exit code 0
```

Screenshot:



****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 2 **Date of Implementation:** 19. 5. 23

****

Q4. Write a program which will accept file name from command line and will show the contents of the file.

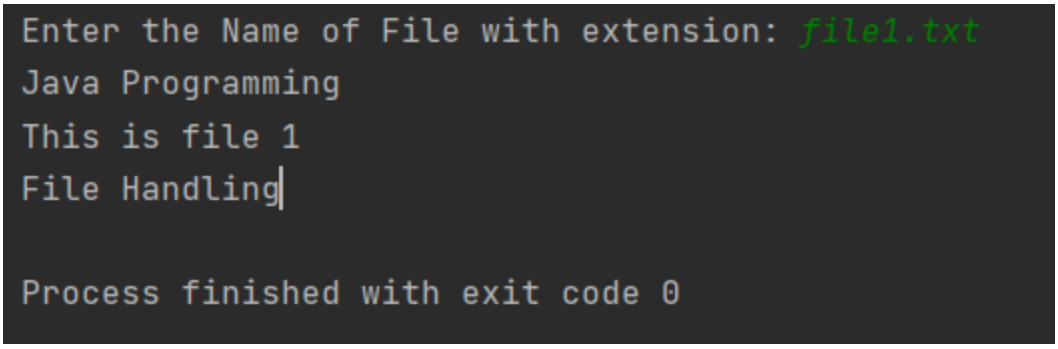
Program:

```
import java.util.Scanner;
import java.io.*;

public class CmdLineFileRead
{
   public static void main(String[] args)
   {
      String fname;
      Scanner scan = new Scanner(System.in);

      System.out.print("Enter the Name of File with extension: ");
      fname = scan.nextLine();

      String line = null;
      try
      {
         FileReader fileReader = new FileReader(fname);

         BufferedReader bufferedReader = new BufferedReader(fileReader);

         while((line = bufferedReader.readLine()) != null)
         {
            System.out.println(line);
         }
         bufferedReader.close();
      }
      catch(IOException ex)
      {
         System.out.println("\nError occurred");
         System.out.println("Exception Name: " +ex);
      }
   }
}
```

Output:

```
Enter the Name of File with extension: file1.txt
Java Programming
This is file 1
File Handling

Process finished with exit code 0
```

Screenshot:



****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 2 **Date of Implementation:** 19. 5. 23

****

Q.5 Write a Java program to concatenate multiple text files sequentially into single file. Accept file names from command line arguments.

Program:

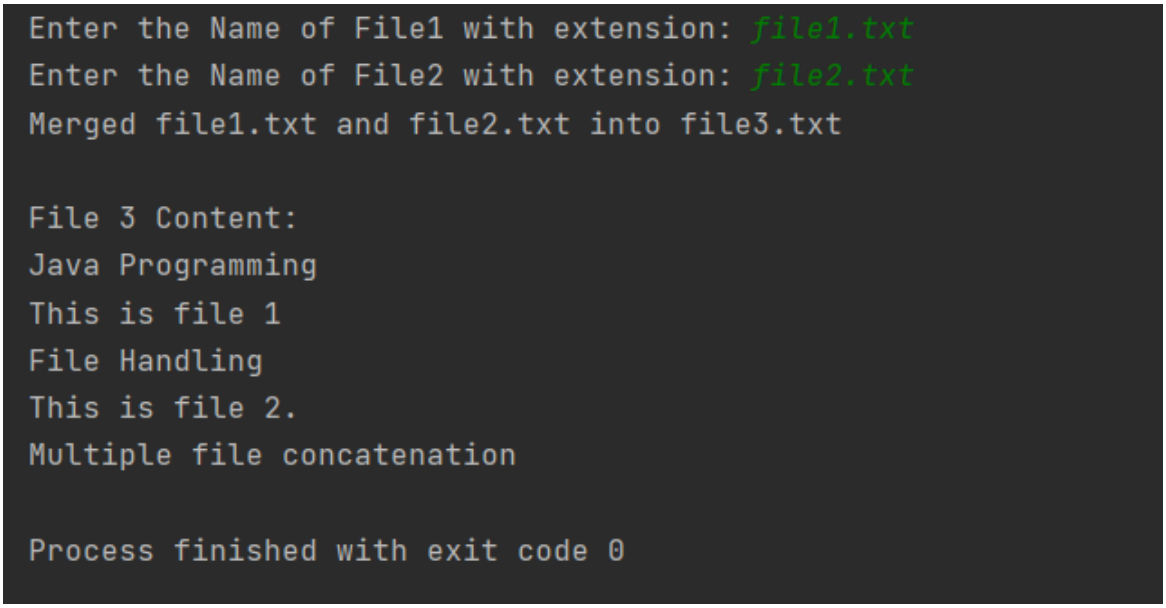
```
import java.io.*;
import java.util.*;

public class MultiFileConcat
{
   public static void main(String[] args) throws IOException    {
      String f1,f2;
      Scanner scan = new Scanner(System.in);

      System.out.print("Enter the Name of File1 with extension: ");
      f1 = scan.nextLine();

      System.out.print("Enter the Name of File2 with extension: ");
      f2 = scan.nextLine();


      PrintWriter pw = new PrintWriter("file3.txt");

      BufferedReader br = new BufferedReader(new FileReader(f1));

      String line = br.readLine();

      while (line != null)       {
         pw.println(line);
         line = br.readLine();
      }

      br = new BufferedReader(new FileReader(f2));
      line = br.readLine();

      while(line != null)       {
         pw.println(line);
         line = br.readLine();
      }

      pw.flush();
      br.close();
      pw.close();

      System.out.println("Merged file1.txt and file2.txt into file3.txt\n");

      BufferedReader bufferedReader = new BufferedReader(new FileReader("file3.txt"));
```

```
        System.out.println("File 3 Content: ");
        while((line = bufferedReader.readLine()) != null)       {
            System.out.println(line);
        }
        bufferedReader.close();
    }
}
```

Output:

```
Enter the Name of File1 with extension: file1.txt
Enter the Name of File2 with extension: file2.txt
Merged file1.txt and file2.txt into file3.txt

File 3 Content:
Java Programming
This is file 1
File Handling
This is file 2.
Multiple file concatenation

Process finished with exit code 0
```

Screenshot:



****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 3 **Date of Implementation:** 6. 6. 23

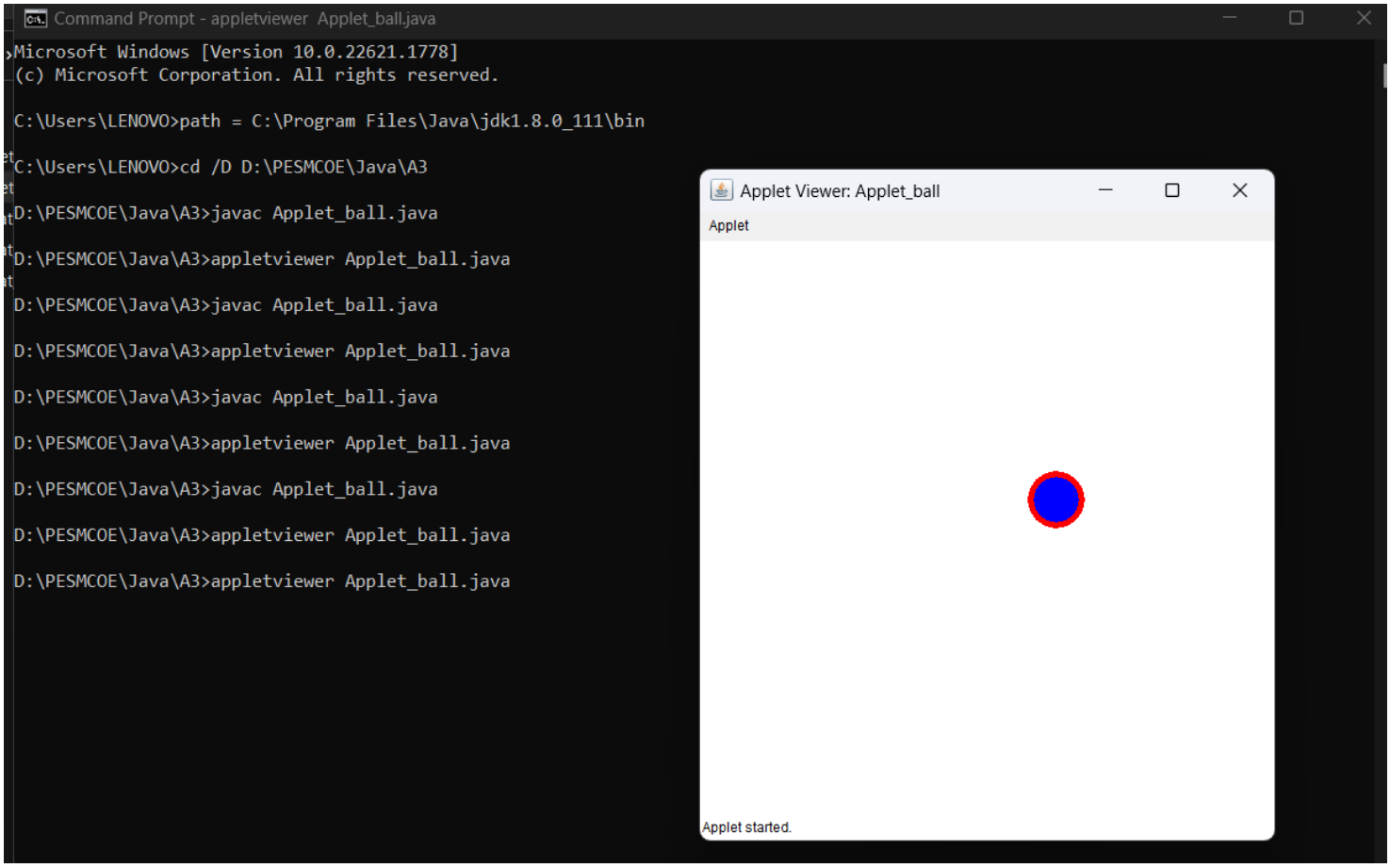
****

Q1. Write an applet to display scrolling ball in an applet window.

Program:

```
import java.awt.Color;
import java.awt.Graphics;
import javax.swing.JApplet;
/*<applet code="Applet_ball" width=500 height=500></applet>*/

public class Applet_ball extends JApplet implements Runnable{

   int x=0, increment =1;
   Thread t=null;

   public void start() {
      t = new Thread(this);
      t.start();
   }
   public void run() {
      while(true){
         if(x==0) {
            increment=1;
         }
         else if(x > getWidth() -50)  {
            increment=-1;
         }
         x+=increment;
         repaint();
        try{
         Thread.sleep(10);
        }catch(Exception e){
           System.out.println(e);
         }
      }
   }
   public void paint(Graphics g)
   {
      g.setColor(Color.white);
      g.fillRect(0, 0, 500, 500);
      g.setColor(Color.red);
      g.fillOval(x, 200, 50, 50);
      g.setColor(Color.blue);
      g.fillOval(x+5, 205, 40, 40);
   }

}
```

Output:


Command Prompt - appletviewer Applet_ball.java
Microsoft Windows [Version 10.0.22621.1778]
(c) Microsoft Corporation. All rights reserved.
C:\Users\LENOVO>path = C:\Program Files\Java\jdk1.8.0_111\bin
C:\Users\LENOVO>cd /D D:\PESMCOE\Java\A3
D:\PESMCOE\Java\A3>javac Applet_ball.java
D:\PESMCOE\Java\A3>appletviewer Applet_ball.java
D:\PESMCOE\Java\A3>javac Applet_ball.java
D:\PESMCOE\Java\A3>appletviewer Applet_ball.java
D:\PESMCOE\Java\A3>javac Applet_ball.java
D:\PESMCOE\Java\A3>appletviewer Applet_ball.java
D:\PESMCOE\Java\A3>javac Applet_ball.java
D:\PESMCOE\Java\A3>appletviewer Applet_ball.java
D:\PESMCOE\Java\A3>appletviewer Applet_ball.java
Applet Viewer: Applet_ball
Applet
Applet started.

Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2022-23

(310919) Java Programming Laboratory

****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 3 **Date of Implementation:** 6. 6. 23

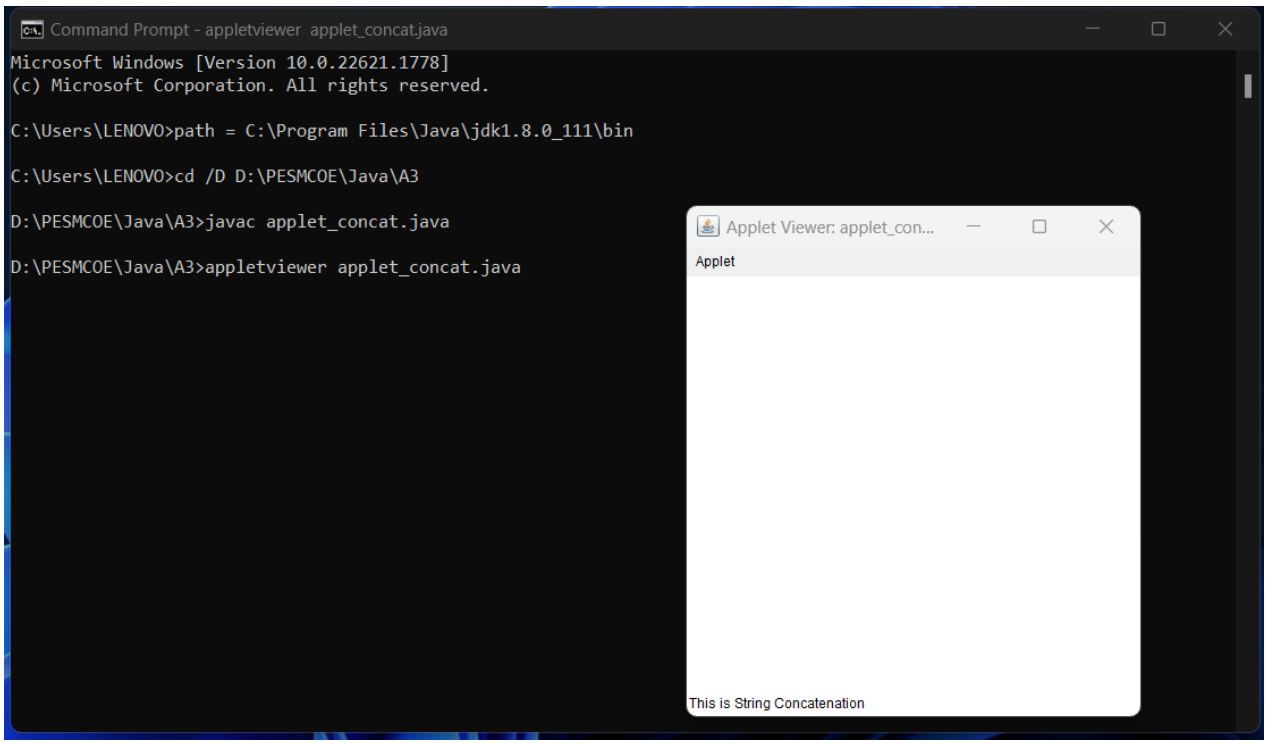
****

Q2. Write an applet program that accepts two input string using <param> tag and concatenate the strings and display it in window.

Program:

```
import java.applet.*;
import java.awt.*;
 /*<applet code = applet_concat.class height = 400 width = 400>
        <param name = "string1" value = "This is">
        <param name = "string2" value = " String Concatenation">
 </applet>*/
 public class applet_concat extends Applet
 {
        String str1;
        public void init() {
            str1 = getParameter("string1").concat(getParameter("string2"));
        }
        public void paint(Graphics g) {
            showStatus(str1);
        }
 }
```

Output:



****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 3 **Date of Implementation:** 6. 6. 23

****

Q3. Write a program to design an admission enquiry form using Swing.

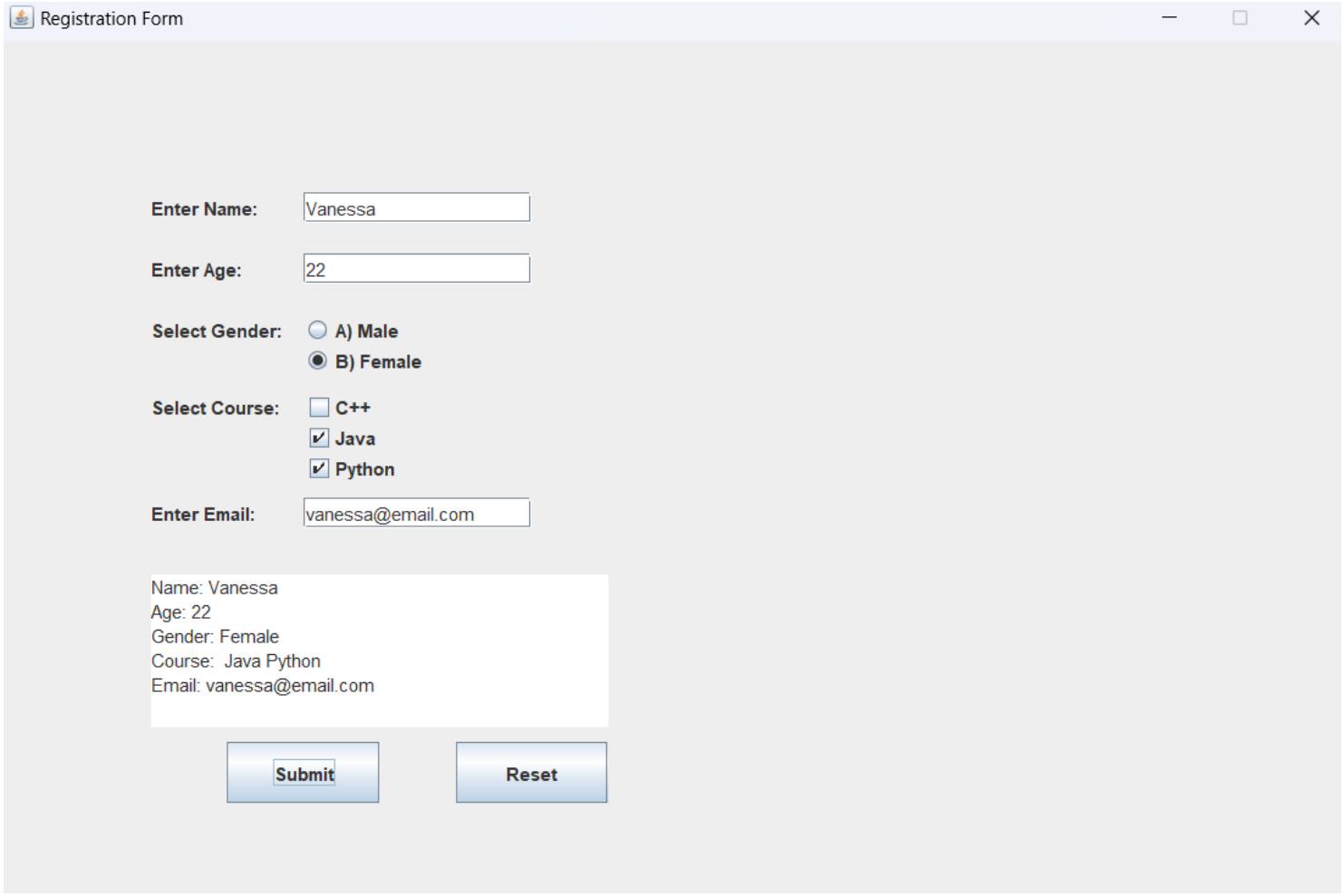
Program:

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

class MyFrame extends JFrame implements ActionListener{

   Container f = new Container();
   JLabel lname = new JLabel("Enter Name: ");
   JTextField tname = new JTextField();
   JLabel lage = new JLabel("Enter Age:");
   JTextField tage = new JTextField();
   JLabel lgender = new JLabel("Select Gender: ");
   JRadioButton r1=new JRadioButton("A) Male");
   JRadioButton r2=new JRadioButton("B) Female");
   ButtonGroup bg=new ButtonGroup();
   JLabel lcourse = new JLabel("Select Course: ");
   JCheckBox checkBox1 = new JCheckBox("C++");
   JCheckBox checkBox2 = new JCheckBox("Java");
   JCheckBox checkBox3 = new JCheckBox("Python");
   JLabel lemail = new JLabel("Enter Email:");
   JTextField temail = new JTextField();
   JTextArea info = new JTextArea();
   JButton b=new JButton("Submit");
   JButton r=new JButton("Reset");
   public MyFrame()
   {
      setTitle("Registration Form");
      setBounds(300, 90, 900, 600);
      setDefaultCloseOperation(EXIT_ON_CLOSE);
      setResizable(false);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      f = getContentPane();
      f.setLayout(null);

      lname.setBounds(100,100,100,20);
      tname.setBounds(200,100,150,20);
      lage.setBounds(100,140,100,20);
      tage.setBounds(200,140,150,20);
```

```
    lgender.setBounds(100,180,100,20);
    r1.setBounds(200,180, 100,20);
    r2.setBounds(200,200, 100,20);
    bg.add(r1);
    bg.add(r2);


    lcourse.setBounds(100, 230,100,20);
    checkBox1.setBounds(200,230, 100,20);
    checkBox2.setBounds(200,250, 100,20);
    checkBox3.setBounds(200,270, 100,20);

    lemail.setBounds(100,300,100,20);
    temail.setBounds(200,300,150,20);
    info.setBounds(100, 350,300,100);


    b.setBounds(150,460,100, 40);//x axis, y axis, width, height
    b.addActionListener(this);

    r.setBounds(300,460,100, 40);//x axis, y axis, width, height
    r.addActionListener(this);

    f.add(b);//adding button in JFrame
    f.add(r);
    f.add(lname);
    f.add(tname);
    f.add(lage);
    f.add(tage);
    f.add(lgender);
    f.add(r1);
    f.add(r2);
    f.add(lcourse);
    f.add(checkBox1);
    f.add(checkBox2);
    f.add(checkBox3);
    f.add(lemail);
    f.add(temail);
    f.add(info);
    f.setSize(600,700);//400 width and 500 height
   f.setLayout(null);//using no layout managers
    setVisible(true);//making the frame visible
}

@Override
public void actionPerformed(ActionEvent e) {

    if (e.getSource()==b){
        String name, age, gender, course, email;

        name = tname.getText();
        age = tage.getText();
```

```
            if (r1.isSelected()){
               gender = "Male";
            }else{
               gender = "Female";
            }

            course = "";
            if (checkBox1.isSelected()){
               course +=" " + checkBox1.getText();
            }

            if(checkBox2.isSelected()){
               course += " " + checkBox2.getText();
            }

            if(checkBox3.isSelected()){
               course += " " + checkBox3.getText();
            }

            email = temail.getText();

            info.setText("Name: " + name + "\nAge: "+ age + "\nGender: "+ gender+ "\nCourse:
"+course + "\nEmail: "+ email);

         }

         if (e.getSource()==r){
            String def = "";
            tname.setText(def);
            tage.setText(def);
            temail.setText(def);
            r1.setSelected(false);
            r2.setSelected(false);
            checkBox1.setSelected(false);
            checkBox2.setSelected(false);
            checkBox3.setSelected(false);
            info.setText(def);
         }

      }
   }

   class formSwing {
      public static void main(String[] args) {
         MyFrame f=new MyFrame();//creating instance of JFrame
      }

   }
```

Output:


Registration Form
Enter Name:
Vanessa
Enter Age:
22
Select Gender:
A) Male
B) Female
Select Course:
C++
Java
Python
Enter Email:
vanessa@email.com
Name: Vanessa
Age: 22
Gender: Female
Course: Java Python
Email: vanessa@email.com
Submit
Reset

****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 3 **Date of Implementation:** 6. 6. 23

****

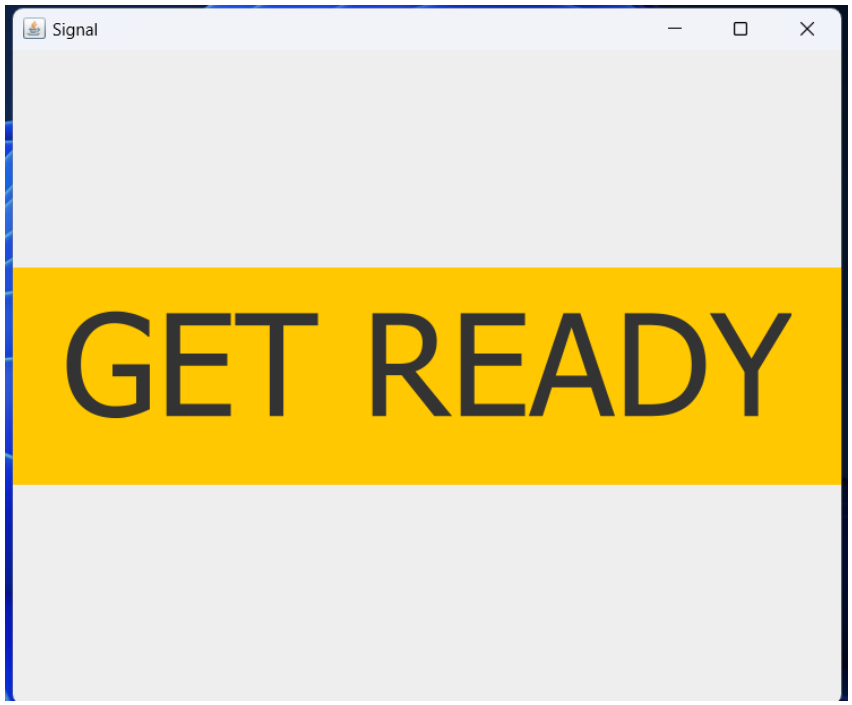
Q4. Design a simple application of Traffic Signal using AWT & Multithreading in java.

Program:

```
import javax.swing.*;
import java.awt.*;
public class TrafficSignal extends JFrame {

   public TrafficSignal(){

      Font font = new Font("Tahoma",100,100);
      GridLayout layout = new GridLayout(3,1,0,0);
      setLayout(layout);
      setTitle("Signal");
      setBounds(200,200,600,500);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setVisible(true);

      JPanel red = new JPanel();
      red.setBackground(Color.RED);
      JLabel stop = new JLabel("STOP");
      stop.setFont(font);
      red.add(stop);
      add(red);
      red.setVisible(false);

      JPanel orange = new JPanel();
      orange.setBackground(Color.ORANGE);
      JLabel ready = new JLabel("GET READY");
      ready.setFont(font);
      orange.add(ready);
      add(orange);
      orange.setVisible(false);

      JPanel green = new JPanel();
      green.setBackground(Color.GREEN);
      JLabel go = new JLabel("GO");
      go.setFont(font);
      green.add(go);
      add(green);
      green.setVisible(false);
```

```
        Thread t = new Thread();
        t.start();

        for(;;){
           try {
              red.setVisible(true);

              t.sleep(1000);
              red.setVisible(false);

              orange.setVisible(true);
              t.sleep(1000);
              orange.setVisible(false);

              green.setVisible(true);
              t.sleep(1000);
              green.setVisible(false);

              orange.setVisible(true);
              t.sleep(1000);
              orange.setVisible(false);

           } catch (InterruptedException e) {
              e.printStackTrace();
           }
        }
     }

     public static void main(String [] args){
        new TrafficSignal();
     }
  }
```

Output:



****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 3 **Date of Implementation:** 6. 6. 23

****

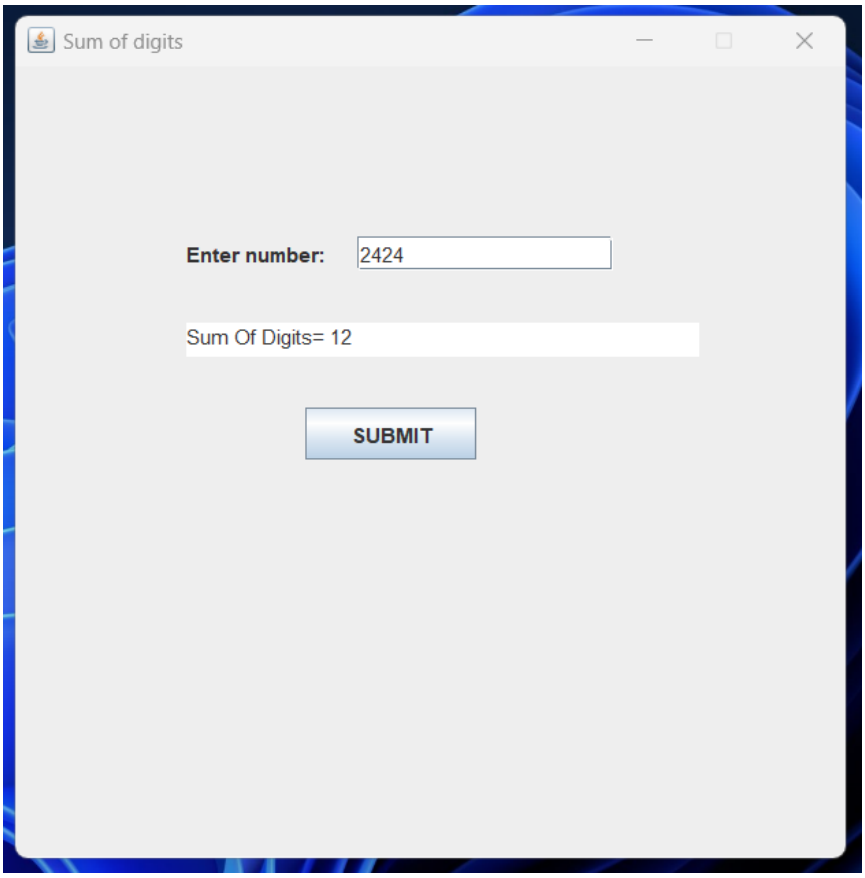
Q5. Create an application that receives a number through a JTextField and print the sum of the individual digits when the submit button is pressed.

Program:

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

class sum extends JFrame implements ActionListener {

   JLabel l = new JLabel("Enter number: ");
   JTextField text = new JTextField();
   JTextArea msg = new JTextArea();
   JButton sub = new JButton("SUBMIT");
   Container c = new Container();

   public sum(){
      setTitle("Sum of digits");
      setBounds(300, 90, 500, 500);
      setDefaultCloseOperation(EXIT_ON_CLOSE);
      setResizable(false);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      c = getContentPane();

      l.setBounds(100,100,100,20);
      text.setBounds(200,100,150,20);
      msg.setBounds(100, 150, 300, 20);
      sub.setBounds(170,200,100,30);

      c.add(l);
      c.add(text);
      c.add(msg);
      c.add(sub);
      sub.addActionListener(this);

      c.setSize(300,300);//400 width and 500 height
      c.setLayout(null);//using no layout managers
      setVisible(true);//making the frame visible
   }
```

```
@Override
   public void actionPerformed(ActionEvent e) {
      if(e.getSource()==sub){

         int n = Integer.parseInt(text.getText());

         int r,s =0;

         while(n>0){
            r = n % 10;
            s += r;
            n = n / 10;
         }

         msg.setText("Sum Of Digits= "+ String.valueOf(s));
      }
   }
}

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

      sum  f=new sum();
   }

}
```

Output:



****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 4 **Date of Implementation:** 13. 6. 23

****

Q1. Design GUI based JDBC application to navigate (first, last, next, previous) through student records. (Assume suitable table structure).

Program:

```
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.SwingUtilities;

public class Student extends JFrame {

	private Connection con;
	private Statement stmt;
	private ResultSet rs;
	private JTextField txtName;
	private JTextField txtID;
	private JTextField txtAddress;

	public Student() {
		setTitle("Student Records");
		setSize(400,200);
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setLocationRelativeTo(null);
		setLayout(new GridLayout(5, 2));
		JLabel lblName = new JLabel("Name:");
		txtName = new JTextField(20);
		JLabel lblID = new JLabel("ID:");
		txtID = new JTextField(10);
		JLabel lblAddress = new JLabel("Address:");
		txtAddress = new JTextField(20);
		JButton btnFirst = new JButton("First");
		JButton btnLast = new JButton("Last");
```

```
JButton btnNext = new JButton("Next");
JButton btnPrevious = new JButton("Previous");

btnFirst.addActionListener(new ActionListener() {

	@Override
	public void actionPerformed(ActionEvent e) {
		// TODO Auto-generated method stub
		showFirstRecord();
	}

});
btnLast.addActionListener(new ActionListener() {

	@Override
	public void actionPerformed(ActionEvent e) {
		// TODO Auto-generated method stub
		showLastRecord();
	}
});

btnNext.addActionListener(new ActionListener() {

	@Override
	public void actionPerformed(ActionEvent e) {
		// TODO Auto-generated method stub
		showNextRecord();
	}
});

btnPrevious.addActionListener(new ActionListener() {

	@Override
	public void actionPerformed(ActionEvent e) {
		// TODO Auto-generated method stub
		showPreviousRecord();
	}
});

add(lblName);
add(txtName);
add(lblID);
add(txtID);
add(lblAddress);
add(txtAddress);
add(btnFirst);
add(btnLast);
add(btnNext);
add(btnPrevious);
```

```
	try {
			Class.forName("oracle.jdbc.driver.OracleDriver");
			String url = "jdbc:oracle:thin:@localhost:1521:XE";
			String username = "system";
			String password = "oracle";

			con = DriverManager.getConnection(url,username,password);
			stmt = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE);
			rs = stmt.executeQuery("SELECT * FROM jdbc_students");


		} catch (Exception e) {
			e.printStackTrace();
		}

		showFirstRecord();
		setVisible(true);
	}

	protected void showLastRecord() {
		try {
			if(rs.last()) {
				displayCurrentRecord();
			}
		} catch (SQLException e) {
			e.printStackTrace();
		}
	}

	protected void showFirstRecord() {
		try {
			if(rs.first()) {
				displayCurrentRecord();
			}
		} catch (SQLException e) {
			// TODO: handle exception
			e.printStackTrace();
		}
	}

	private void showPreviousRecord() {
		try {
			if(rs.previous()) {
				displayCurrentRecord();
			}else {
				rs.next();
				JOptionPane.showMessageDialog(null, "Start of records");
			}

		} catch (SQLException e) {
			e.printStackTrace();
```

```
		}
	}

	private void showNextRecord() {
		try {
			if(rs.next()) {
				displayCurrentRecord();
			}else {
				rs.previous();
				JOptionPane.showMessageDialog(null, "End of records");
			}

		} catch (SQLException e) {
			e.printStackTrace();

		}
	}
	private void displayCurrentRecord() {
		try {
			String name = rs.getString("name");
			int id = rs.getInt("id");
			String address = rs.getString("addr");
			txtName.setText(name);
			txtID.setText(String.valueOf(id));
			txtAddress.setText(address);

		} catch (SQLException e) {
			e.printStackTrace();
		}
	}
	public static void main(String[] args) {
		SwingUtilities.invokeLater(new Runnable() {
			public void run() {
				new Student();
			}
		});
	}
}
```

Screenshot:

Results Explain Describe Saved SQL History

ID	NAME	ADDR
10	Vanessa	Pune
20	Bhushan	Pune
30	Ronak	Lonavla
40	Tanima	Kolkata
50	Shruti	Lucknow
60	Aman	Gujurat
70	Sakshi	Pune
80	Mahesh	Odissa

8 rows returned in 0.09 seconds CSV Export

****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 4 **Date of Implementation:** 13. 6. 23

****

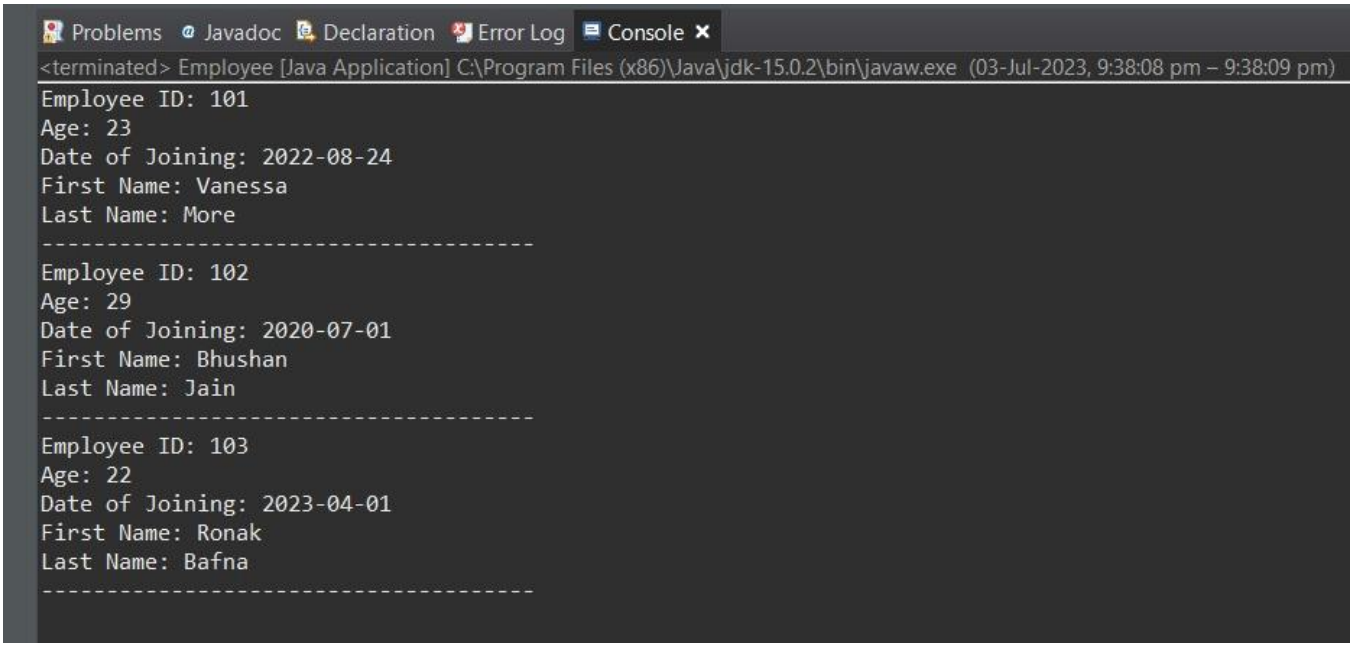
Q2. Write a Java program to display the employee id, age, Date of joining, first name and last name using JDBC connectivity. (Assume suitable table structure)

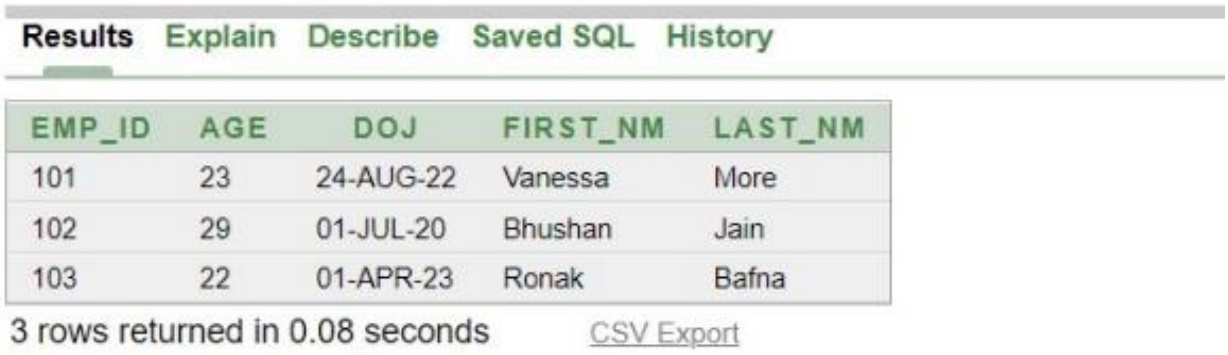
Program:

```
import java.sql.Connection;
import java.sql.Date;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class Employee {

    public static void main(String[] args) {
        try {

            Class.forName("oracle.jdbc.driver.OracleDriver");
            String url = "jdbc:oracle:thin:@localhost:1521:XE";
            String username = "system";
            String password = "oracle";
            Connection connection = DriverManager.getConnection(url, username, password);
            Statement statement = connection.createStatement();

            String query =  "SELECT emp_id, age, doj, first_nm, last_nm FROM jdbc_emp";
            ResultSet resultSet = statement.executeQuery(query);

            while(resultSet.next()) {
                int employeeId = resultSet.getInt("emp_id");
                int age = resultSet.getInt("age");
                Date joinDate = resultSet.getDate("doj");
                String firstName = resultSet.getString("first_nm");
                String lastName = resultSet.getString("last_nm");

                System.out.println("Employee ID: " + employeeId);
                System.out.println("Age: " + age);
                System.out.println("Date of Joining: " + joinDate);
                System.out.println("First Name: " + firstName);
                System.out.println("Last Name: " + lastName);
                System.out.println("-----------------------------------");
            }

            resultSet.close();
            statement.close();
```

```
                connection.close();

            } catch (ClassNotFoundException | SQLException e) {
                e.printStackTrace();
            }
        }
    }
```

Screenshot:





****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 4 **Date of Implementation:** 13. 6. 23

****

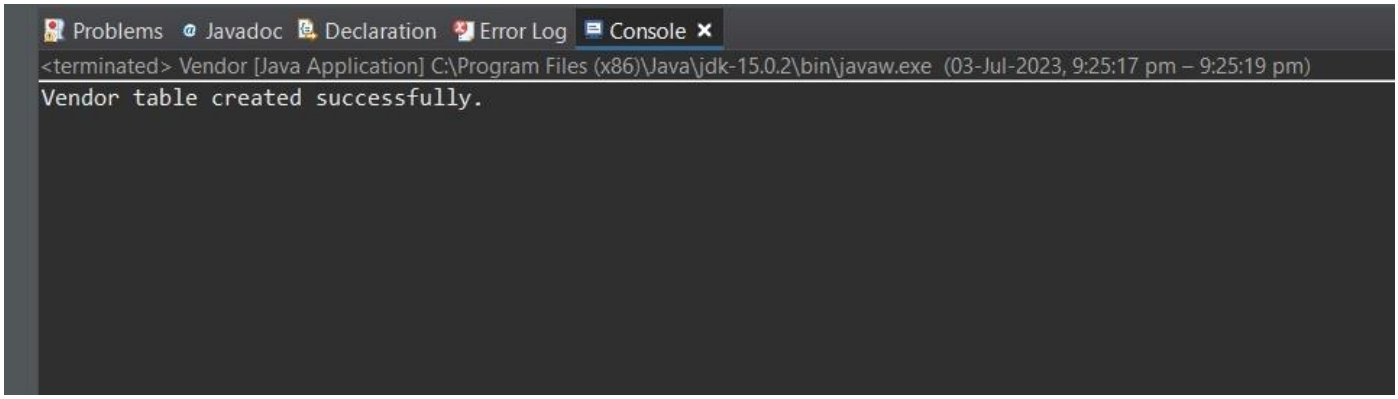
Q3. Oracle database is stored on server.

a. Write a java program to create vendor table with vendor no, name, balance Amount

Program:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

public class Vendor {

	public static void main(String[] args) {
		try{

			Class.forName("oracle.jdbc.driver.OracleDriver");
			String url = "jdbc:oracle:thin:@localhost:1521:XE";
			String username = "system";
			String password = "oracle";
			Connection connection = DriverManager.getConnection(url, username, password);
			Statement statement = connection.createStatement();

			String createTableQuery = "create table vendor(vendor_no int primary key,name
varchar(20),balance_amount float)";
			statement.executeUpdate(createTableQuery);

			System.out.println("Vendor table created successfully.");
			statement.close();
			connection.close();
		}
		catch (ClassNotFoundException | SQLException e){
			e.printStackTrace();
		}

	}

}
```

Screenshot:

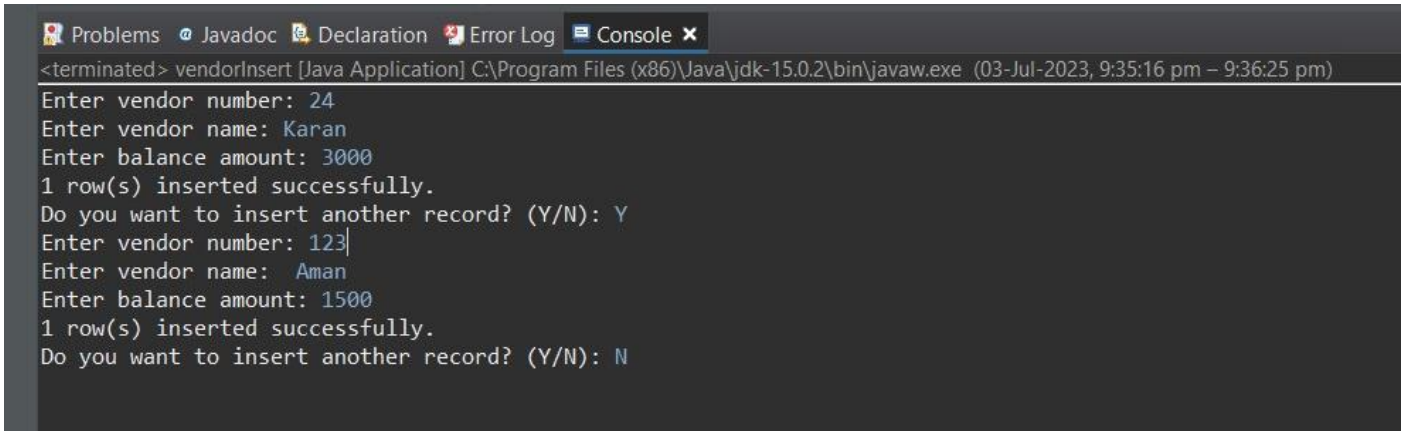

Problems
Javadoc
Declaration
Error Log
Console
<terminated> Vendor [Java Application] C:\Program Files (x86)\Java\jdk-15.0.2\bin\javaw.exe (03-Jul-2023, 9:25:17 pm – 9:25:19 pm)
Vendor table created successfully.

b. Write java program to insert rows in above-mentioned table. Accept Input from user.

Program:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.Scanner;
public class vendorInsert {
	public static void main(String[] args) {
		try
		{
			Class.forName("oracle.jdbc.driver.OracleDriver");
			String url = "jdbc:oracle:thin:@localhost:1521:XE";
			String username = "system";
			String password = "oracle";
			Connection connection = DriverManager.getConnection(url, username, password);

			String insertQuery = "insert into vendor(vendor_no, name, balance_amount)
VALUES (?, ?, ?)";
			PreparedStatement statement = connection.prepareStatement(insertQuery);
			Scanner scanner = new Scanner(System.in);
			String userInput;

			do{
				System.out.print("Enter vendor number: ");
				int vendorNo = scanner.nextInt();
				scanner.nextLine();
				System.out.print("Enter vendor name: ");
				String name = scanner.nextLine();
				System.out.print("Enter balance amount: ");
				double balanceAmount = scanner.nextDouble();

				statement.setInt(1, vendorNo);
				statement.setString(2, name);
				statement.setDouble(3, balanceAmount);

				int rowsInserted = statement.executeUpdate();
				System.out.println(rowsInserted + " row(s) inserted successfully.");

				System.out.print("Do you want to insert another record? (Y/N): ");
				userInput = scanner.next().toUpperCase();
			} while (!userInput.equals("N"));

			statement.close();
			connection.close();
		}
		catch (ClassNotFoundException | SQLException e) {
			e.printStackTrace();
		}
	}
}
```

Screenshot:


Problems
Javadoc
Declaration
Error Log
Console
<terminated> vendorInsert [Java Application] C:\Program Files (x86)\Java\jdk-15.0.2\bin\javaw.exe (03-Jul-2023, 9:35:16 pm – 9:36:25 pm)
Enter vendor number: 24
Enter vendor name: Karan
Enter balance amount: 3000
1 row(s) inserted successfully.
Do you want to insert another record? (Y/N): Y
Enter vendor number: 123
Enter vendor name: Aman
Enter balance amount: 1500
1 row(s) inserted successfully.
Do you want to insert another record? (Y/N): N

****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 5 **Date of Implementation:** 27. 6. 23

****

Q1. Write a program to establish connection between client and server.

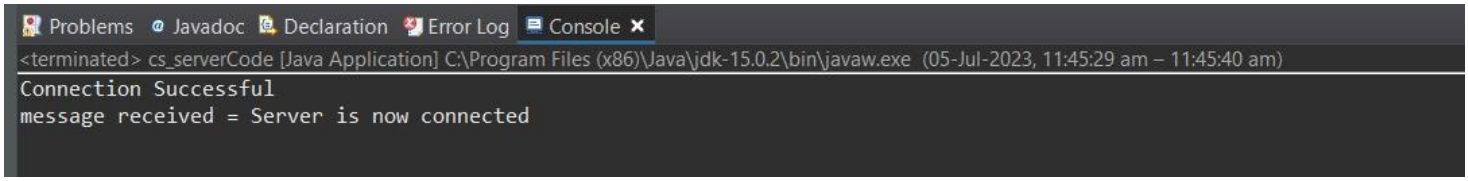
Server Program:

```
import java.io.*;
import java.net.*;
public class cs_serverCode{
    public static void main(String[] args) {
        try
        {
            ServerSocket sos=new ServerSocket(6666);
            Socket soc= sos.accept();
            ObjectInputStream ois=new
            ObjectInputStream(soc.getInputStream());
            String str=(String)ois.readUTF();
            System.out.println("Connection Successful\nmessage received = "+str);
            sos.close();
        }
        catch(Exception e)          {
            System.out.println(e);
        }
    }
}
```

Client Program:

```
import java.io.*;
import java.net.*;
public class cs_clientCode{
    public static void main(String[] args) {
        try             {
            Socket soc = new Socket("localhost",6666);
            ObjectOutputStream out= new
            ObjectOutputStream(soc.getOutputStream());
            out.writeUTF("Server is now connected");
            out.flush();
            soc.close();
        }
        catch(Exception e){
            System.out.println(e);
        }
    }
}
```

Screenshot:


Problems
Javadoc
Declaration
Error Log
Console
<terminated> cs_serverCode [Java Application] C:\Program Files (x86)\Java\jdk-15.0.2\bin\javaw.exe (05-Jul-2023, 11:45:29 am – 11:45:40 am)
Connection Successful
message received = Server is now connected

****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 5 **Date of Implementation:** 27. 6. 23

****

Q2. Write a Java Socket program for client server chatting application.

Server Program:

```
import java.io.BufferedReader;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.net.ServerSocket;
import java.net.Socket;
public class chat_server
{
	public static void main(String[] args)
	{
		DataInputStream din = null;
		ServerSocket serverSocket = null;
		DataOutputStream dout = null;
		BufferedReader br = null;
		try
		{
			serverSocket = new ServerSocket(6655);
			System.out.println("Server is Waiting for client request... ");
			Socket socket = serverSocket.accept();
			din = new DataInputStream(socket.getInputStream());
			OutputStream outputStream = socket.getOutputStream();
			dout = new DataOutputStream(outputStream);
			br = new BufferedReader(new InputStreamReader(System.in));
			String strFromClient = "", strToClient = "";

			while (!strFromClient.equals("stop"))
			{
				strFromClient = din.readUTF();
				System.out.println("CLIENT: " + strFromClient);
				strToClient = br.readLine();
				dout.writeUTF(strToClient);
				dout.flush();
			}
		}
```

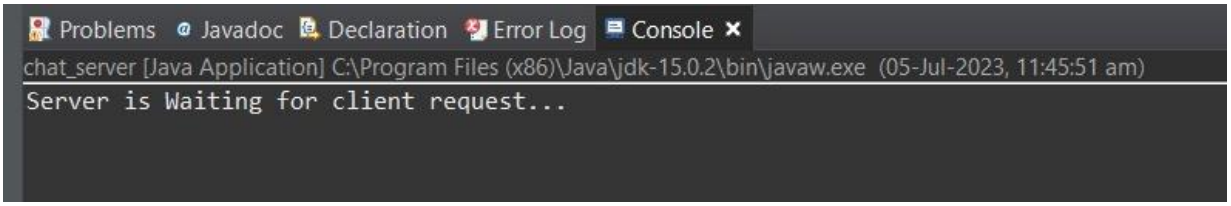
```
                catch (Exception exe)
                {
                        exe.printStackTrace();
                }
                finally
                {
                        try
                        {
                                if (br != null)
                                {
                                        br.close();
                                }
                                if (din != null)
                                {
                                        din.close();
                                }
                                if (dout != null)
                                {
                                        dout.close();
                                }
                                if (serverSocket != null)
                                {
                                        serverSocket.close();
                                }
                        }
                        catch (IOException e)
                        {
                                e.printStackTrace();
                        }
                }
        }
}
```

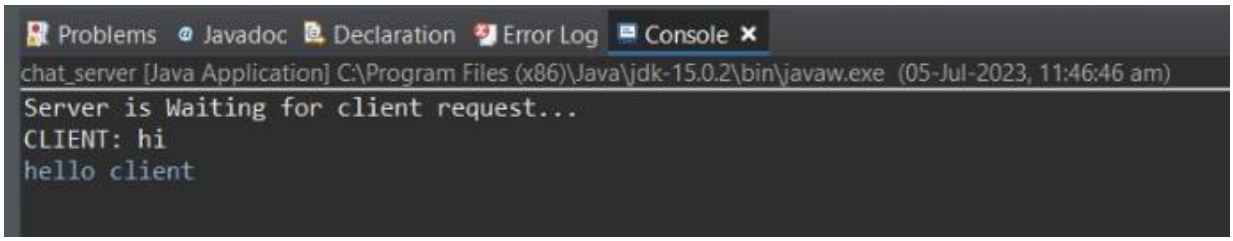
Client Program:

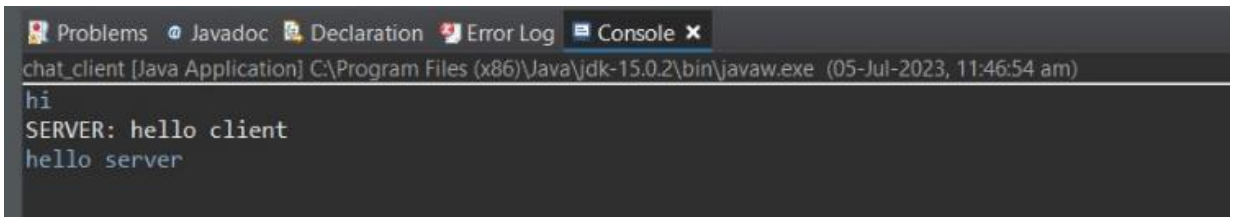
```
import java.io.BufferedReader;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.net.Socket;
public class chat_client
{
        public static void main(String[] args)
        {
                Socket socket = null;
                DataInputStream din = null;
                DataOutputStream dout = null;
                BufferedReader br = null;
        try
        {
                        socket = new Socket("localhost", 6655);
```

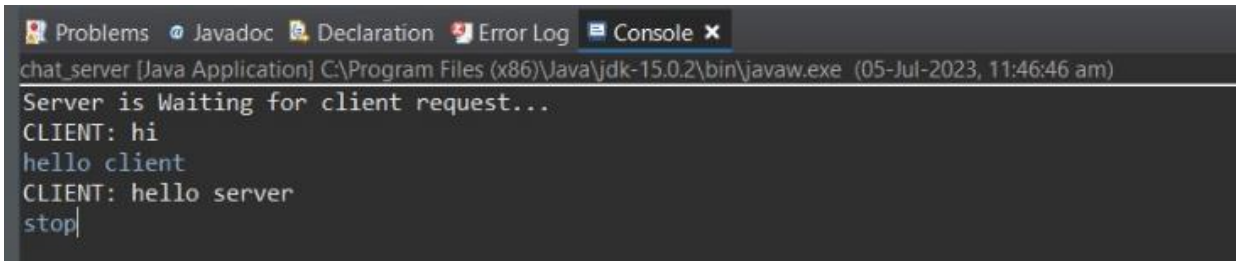
```
                din = new DataInputStream(socket.getInputStream());
                OutputStream outputStream = socket.getOutputStream();
                dout = new DataOutputStream(outputStream);
                br = new BufferedReader(new InputStreamReader(System.in));
                String strFromServer = "", strToClient = "";
            while (!strFromServer.equals("stop"))
            {
                strFromServer = br.readLine();
                dout.writeUTF(strFromServer);
                dout.flush();
                strToClient = din.readUTF();
                System.out.println("SERVER: " + strToClient);
            }
        }
        catch (Exception exe)
        {
            exe.printStackTrace();
        }
        finally
            {
                try
                {
                    if (br != null)
                    {
                        br.close();
                    }
                    if (din != null)
                    {
                        din.close();
                    }
                    if (dout != null)
                    {
                        dout.close();
                    }
                    if (socket != null)
                    {
                        socket.close();
                    }
                }
                catch (IOException e)
                {
                    e.printStackTrace();
                }
            }
        }
}
```

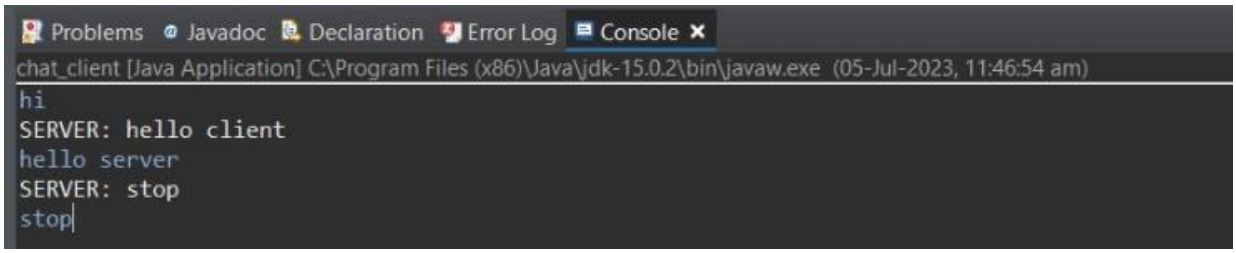
Screenshot:

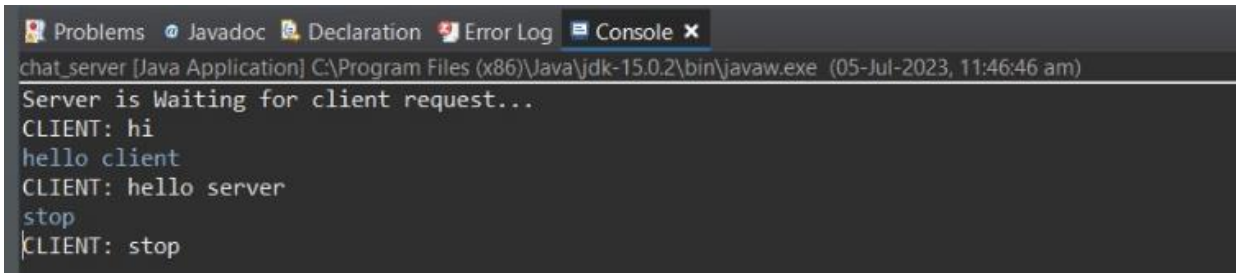

Problems Javadoc Declaration Error Log Console
chat_server [Java Application] C:\Program Files (x86)\Java\jdk-15.0.2\bin\javaw.exe (05-Jul-2023, 11:45:51 am)
Server is Waiting for client request...

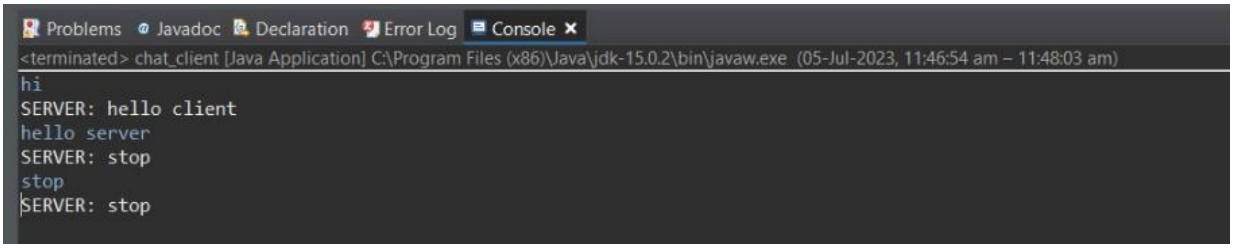

Problems Javadoc Declaration Error Log Console
chat_server [Java Application] C:\Program Files (x86)\Java\jdk-15.0.2\bin\javaw.exe (05-Jul-2023, 11:46:46 am)
Server is Waiting for client request...
CLIENT: hi
hello client


Problems Javadoc Declaration Error Log Console
chat_client [Java Application] C:\Program Files (x86)\Java\jdk-15.0.2\bin\javaw.exe (05-Jul-2023, 11:46:54 am)
hi
SERVER: hello client
hello server


Problems Javadoc Declaration Error Log Console
chat_server [Java Application] C:\Program Files (x86)\Java\jdk-15.0.2\bin\javaw.exe (05-Jul-2023, 11:46:46 am)
Server is Waiting for client request...
CLIENT: hi
hello client
CLIENT: hello server
stop


Problems Javadoc Declaration Error Log Console
chat_client [Java Application] C:\Program Files (x86)\Java\jdk-15.0.2\bin\javaw.exe (05-Jul-2023, 11:46:54 am)
hi
SERVER: hello client
hello server
SERVER: stop
stop


Problems Javadoc Declaration Error Log Console
chat_server [Java Application] C:\Program Files (x86)\Java\jdk-15.0.2\bin\javaw.exe (05-Jul-2023, 11:46:46 am)
Server is Waiting for client request...
CLIENT: hi
hello client
CLIENT: hello server
stop
CLIENT: stop


Problems Javadoc Declaration Error Log Console
<terminated> chat_client [Java Application] C:\Program Files (x86)\Java\jdk-15.0.2\bin\javaw.exe (05-Jul-2023, 11:46:54 am – 11:48:03 am)
hi
SERVER: hello client
hello server
SERVER: stop
stop
SERVER: stop

Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2022-23

(310919) Java Programming Laboratory

****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 5 **Date of Implementation:** 27. 6. 23

****

Q3. Write a program to establish connection between client and server using datagram packet.

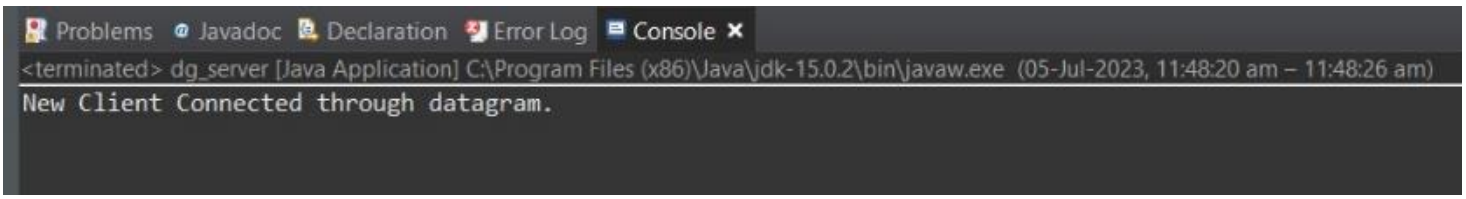
Server Code:

```
import java.net.*;
public class dg_server //receiver
{
	public static void main(String[] args) throws Exception
	{
		DatagramSocket ds = new DatagramSocket(3000);
		byte[] buf = new byte[1024];
		DatagramPacket dp = new DatagramPacket(buf,1024);
		ds.receive(dp);
		String str = new String(dp.getData(), 0,dp.getLength());
		System.out.println(str);
		ds.close();
	}
}
```

Client Code:

```
import java.net.*;
public class dg_client //sender
{
	public static void main(String[] args) throws Exception
	{
		DatagramSocket ds = new DatagramSocket();
		String str = "New Client Connected through datagram.";
		InetAddress ip = InetAddress.getByName("127.0.0.1");
		DatagramPacket dp = new DatagramPacket(str.getBytes(), str.length(), ip, 3000);
		ds.send(dp);
		ds.close();
	}
}
```

Screenshot:



Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2022-23

(310919) Java Programming Laboratory

****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 6 **Date of Implementation:** 7. 7. 23

****

Q1. Write a servlet application to print a Welcome message and the current date and time.

Program:

```
import java.io.IOException;
import java.io.PrintWriter;

import jakarta.servlet.GenericServlet;
import jakarta.servlet.ServletException;
import jakarta.servlet.ServletRequest;
import jakarta.servlet.ServletResponse;
import jakarta.servlet.annotation.WebServlet;
@WebServlet("/welcome")
public class Welcome extends GenericServlet
{
	private static final long serialVersionUID = 1L;
	public void service(ServletRequest req, ServletResponse res) throws IOException,
	ServletException
	{
		//get stream obj
		PrintWriter pw = res.getWriter();
		//write req processing logic
		java.util.Date date = new java.util.Date();
		pw.println("<h1>"+"Welcome"+"</h1>");
		pw.println("<h2>"+"\nCurrent Date & Time: " +date.toString()+"</h2>");
		//close stream object
		pw.close();
	}
}
```

Screenshot:

Welcome

Current Date & Time: Sun Jul 09 23:46:41 IST 2023

****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 6 **Date of Implementation:** 7. 7. 23

****

Q2. Write a Servlet program that accepts the age and name and displays if the user is eligible for voting or not.

Program:

Assign_6_2.html

```
<!DOCTYPE html>
<html>
<head><title>Voting Eligibility Form</title></head>
<body>
<h1>Voting Eligibility Form</h1>
<form action="Assign_6_2" method="post">
<label for="name">Name:</label>
<input type="text" id="name" name="name" required><br><br>
<label for="age">Age:</label>
<input type="number" id="age" name="age" required><br><br>
<input type="submit" value="Check Eligibility">
</form>
</body></html>
```

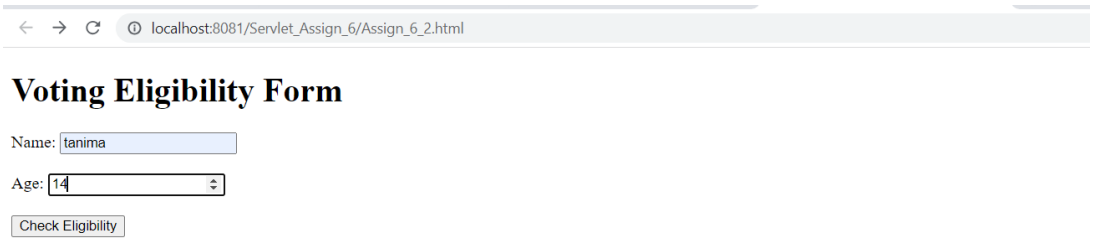
Assign_6_2.java

```
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
public class Assign_6_2 extends HttpServlet
{
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException
{
int age = Integer.parseInt(request.getParameter("age"));
String name = request.getParameter("name");
boolean isEligible = age >= 18;
response.setContentType("text/html");
PrintWriter out = response.getWriter();
out.println("<html>");
out.println("<head><title>Voting Eligibility</title></head>");
out.println("<body>");
out.println("<h1>Voting Eligibility Check</h1>");
out.println("<p>Name: " + name + "</p>");
out.println("<p>Age: " + age + "</p>");
```

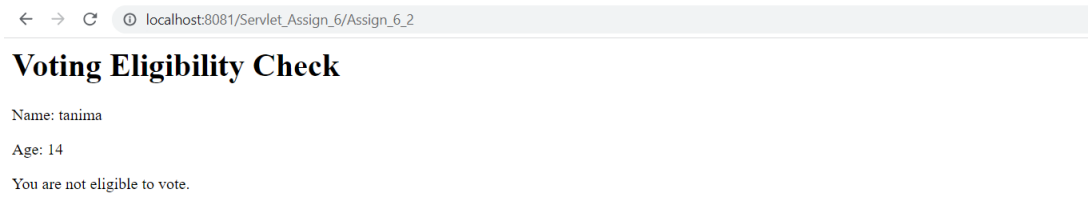
```
if (isEligible)
{
out.println("<p>You are eligible to vote.</p>");
}
else
{
out.println("<p>You are not eligible to vote.</p>");
}
out.println("</body></html>");
}
}
```

Screenshot:

HTML



JAVA



Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2022-23

(310919) Java Programming Laboratory

****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 6 **Date of Implementation:** 7. 7. 23

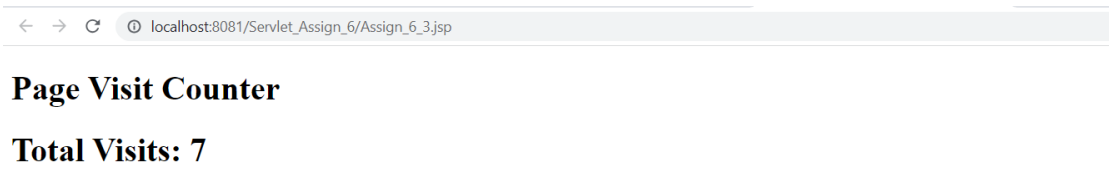
****

Q3. Write a JSP application to count the total number of visits on your website.

Program:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head><title>Page Visit Counter</title></head>
<body>
<h1>Page Visit Counter</h1>
<%
Integer visitCount = (Integer) application.getAttribute("visitCount");
if (visitCount == null)
{
visitCount = 1;
}
else
{
visitCount++;
}
application.setAttribute("visitCount", visitCount);
%>
<h1>Total Visits: <%= visitCount %></h1>
</body></html>
```

Screenshot:



Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2022-23

(310919) Java Programming Laboratory

****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 6 **Date of Implementation:** 7. 7. 23

****

Q4. Write a JSP program to display the grade of a student by accepting the marks of five subjects.

Program:

Assign_6_4.html

```
<!DOCTYPE html>
<html>
<head><title>Student Grade</title></head>
<body>
<h1>Student Grade Form</h1>
<form action="Assign_6_4.jsp" method="post">
<label for="subject1">Computer Networks:</label>
<input type="number" id="subject1" name="subject1" required><br><br>
<label for="subject2">Operating Systems:</label>
<input type="number" id="subject2" name="subject2" required><br><br>
<label for="subject3">Java:</label>
<input type="number" id="subject3" name="subject3" required><br><br>
<label for="subject4">DBMS:</label>
<input type="number" id="subject4" name="subject4" required><br><br>
<label for="subject5">Mobile Computing:</label>
<input type="number" id="subject5" name="subject5" required><br><br>
<input type="submit" value="Calculate Grade">
</form>
</body></html>
```

Assign_6_4.jsp

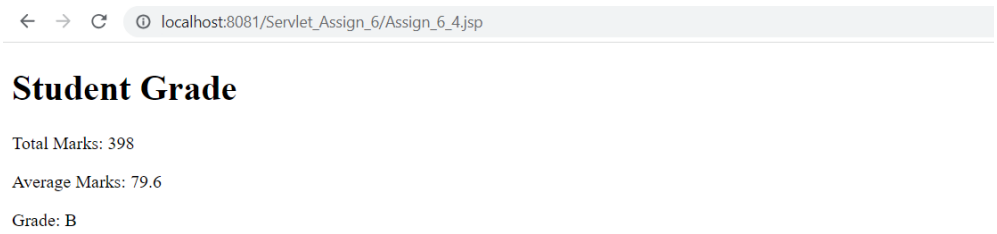
```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head><title>Student Grade</title></head>
<body>
<%
int subject1 = Integer.parseInt(request.getParameter("subject1"));
int subject2 = Integer.parseInt(request.getParameter("subject2"));
int subject3 = Integer.parseInt(request.getParameter("subject3"));
int subject4 = Integer.parseInt(request.getParameter("subject4"));
int subject5 = Integer.parseInt(request.getParameter("subject5"));
int totalMarks = subject1 + subject2 + subject3 + subject4 + subject5;
double averageMarks = totalMarks / 5.0;
String grade;
if (averageMarks >= 90)
{
grade = "A+";
```

```
}
else if (averageMarks >= 80)
{
grade = "A";
}
else if (averageMarks >= 70)
{
grade = "B";
}
else if (averageMarks >= 60)
{
grade = "C";
}
else if (averageMarks >= 50)
{
grade = "D";
}
else
{
grade = "F";
}
%>
<h1>Student Grade</h1>
<p>Total Marks: <%= totalMarks %></p>
<p>Average Marks: <%= averageMarks %></p>
<p>Grade: <%= grade %></p>
</body></html>
```

Screenshot:

HTML

JSP



Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2022-23

(310919) Java Programming Laboratory

****

Class: FY-MCA **Shift / Div:** A **Batch:** F2 **Roll Number:** 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 6 **Date of Implementation:** 7. 7. 23

****

Q5. Write a JSP code for accepting book id from HTML page. Verify it with a book table (Assume suitable fields). If a book already exists then display its details. If not, insert new details into the table.

Program:

Assign_6_5.html

```
<!DOCTYPE html>
<html>
<head>
<title>Book Details Form</title>
</head>
<body>
<h1>Book Details Form</h1>
<form action="Assign_6_5.jsp" method="post">
<label for="bookId">Book ID:</label>
<input type="text" id="bookId" name="bookId" required><br><br>
<input type="submit" value="Submit">
</form>
</body>
</html>
```

Assign_6_5.jsp

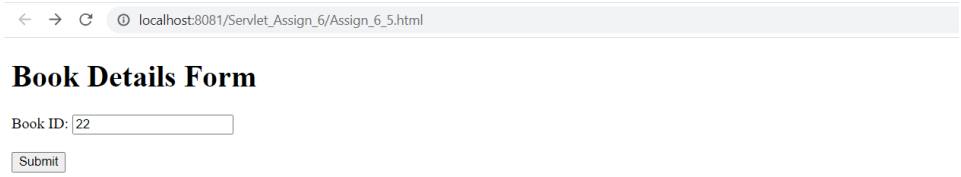
```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ page import="java.sql.*" %>
<!DOCTYPE html>
<html>
<head>
<title>Book Details</title>
</head>
<body>
<%
String bookId = request.getParameter("bookId");
String url = "jdbc:oracle:thin:@localhost:1521:XE";
String username = "system";
String password = "oracle";
Connection conn = null;
PreparedStatement stmt = null;
ResultSet rs = null;
try
{
Class.forName("oracle.jdbc.driver.OracleDriver");
conn = DriverManager.getConnection(url, username, password);
String query = "SELECT * FROM books WHERE id = ?";
```

```
stmt = conn.prepareStatement(query);
stmt.setString(1, bookId);
rs = stmt.executeQuery();
if (rs.next())
{
String bookTitle = rs.getString("title");
String author = rs.getString("author");
int year = rs.getInt("year");
out.println("<h1>Book Details</h1>");
out.println("<p>Book ID: " + bookId + "</p>");
out.println("<p>Title: " + bookTitle + "</p>");
out.println("<p>Author: " + author + "</p>");
out.println("<p>Year: " + year + "</p>");
}
else
{
String insertQuery = "INSERT INTO books (id, title, author, year) VALUES (?, ?, ?, ?)";
stmt = conn.prepareStatement(insertQuery);
stmt.setString(1, bookId);
stmt.setString(2, "New Book");
stmt.setString(3, "Unknown Author");
stmt.setInt(4, 2023);
int rowsAffected = stmt.executeUpdate();
if (rowsAffected > 0)
{
out.println("<h1>Book Details</h1>");
out.println("<p>Book ID: " + bookId + "</p>");
out.println("<p>Title: New Book</p>");
out.println("<p>Author: Unknown Author</p>");
out.println("<p>Year: 2023</p>");
out.println("<p>New book details inserted into the database.</p>");
}
else
{
out.println("<h1>Error</h1>");
out.println("<p>Failed to insert new book details.</p>");
}
}
}
catch (Exception e)
{
e.printStackTrace();
}
finally
{
if (rs != null)
{
try
{
rs.close();
}
catch (SQLException e)
{
e.printStackTrace();
}
}
```

```
if (stmt != null)
{
try
{
stmt.close();
}
catch (SQLException e)
{
e.printStackTrace();
}
}
if (conn != null)
{
try
{
conn.close();
}
catch (SQLException e)
{
e.printStackTrace();
}
}
}
%>
</body>
</html>
```

Screenshot:

HTML



JSP