22/12/2023

Date __/__/__

Page

LAB-1

Q1 write a program to overload the method print that prints sum of 'n' natural numbers when one variable is passed, and prints the prime numbers in a given range when 2 parameters are passed.

```
class Overload
{
  void print (int n)
  {
    int sum = 0;
    for (int i=1; i<=n; i++)
    {
      sum = sum + i;
    }
    System.Out.println ("sum of " + n + " natural nos = "
                        + sum);
  }

void print (int m, int n)
{
  System.Out.println ("prime nos in the range are ");
  for (int i = m; i<=n; i++)
  {
    int flag = 0;
    for (int j = 2; j<= i/2; j++)
    {
      if (i%j == 0)
      {
        flag = 1;
        break;
      }
    }
  }
```

```
if (flag == 0)
System.out.println(i);
}
}
}

class OverloadDemo
{
  public static void main(String[] args)
{
    overload O = new Overload();
      O.print(5);
      O.print(7,13);
  }
}
```

Output

Sum of 5 natural nos = 15
Prime nos in the range are
7
11
13

Q2) Write a program to create a class grocery that has the variables c-name and c-phone. Create a method to accept 3 parameters to specify quantity of dal, quantity of pulses an quantity of sugar. The method to return the total price. Display the name, ph-no and total bill of 3 customers.

Soln

```
class Grocery
{
    string c-name;
    string c-ph;
    double total;
    Grocery ( string c-name, string c-ph)
    {
        this.c-name = c-name;
        this.c-ph = c-ph;
    }

void calc ( double q-dal, double q-pulses, double q-sugar)
{
    total = q-dal*100 + q-pulses * 80 + q-sugar * 50;
}

void display()
{
    System.out.println("Name" + " " + "phone number"
                       + " " + "total");

    System.out.println(c-name + " " + c-ph + " " +
                       total);

    system.out.println();
}
}
```

```
class GRemoh
{
public static void main (String [] args)
 {
   Grocery g1 = new Grocery ("Rama", "9491115111");

   Grocery g2 = new Grocery ("Shama", "7410867891");

   Grocery g3 = new Grocery ("Bhama", "8111188777");

   g1.calc(2,2,1);
   g1.display();
   g2.calc(3,5,2);
   g2.display();
   g3.calc(1,1,0.5);
   g3.display();
 }
}
```

Output

Name	phone number	Total
Rama	8194538611	410.00

Name	phone number	Total
Shania	7110596679	800.00

Name	phone number	Total
Bhama	8104111778	205.0

Date __/__/__

Page _____

Q.3. Write a Java program to calculate roots of a quadratic equation. Use appropriate methods to take inputs, and calculate the roots.

Sol:

```
import java.util.Scanner;
class Quad
{
  int a, b, c;
  double root1, root2, d;
  Scanner s = new Scanner (System.in);

  void input()
  {
    System.out.println("Quadratic equation is in the
    form: ax^2 + bx + c");

    System.out.println("enter a:");
    a = s.nextInt();

    System.out.println("enter b:");
    b = s.nextInt();

    System.out.println("enter c:");
    c = s.nextInt();
  }

  void discriminant()
  {
    d = (b*b) - (4*a*c);
  }
```

```
void calculateRoots()
{
    if (d>0)
{
    System.out.println("roots are real and unequal");

    root1 = (-b + Math.sqrt(d)) / (2*a);

    root2 = (-b * -Math.sqrt(d)) / (2*a);

    System.out.println("First root is:" + root1);

    System.out.println("second root is:" + root2);
}
    else if (d==0)
{
    System.out.println("Roots are real and equal");
    root1 = (-b + Math.sqrt(d)) / (2*a);

    System.out.println("Root:" + root1);
}
else
{
    System.out.println("No real soln, imaginary");

    double real = -b/(2*a);
    double imaginary = Math.sqrt(-d)/(2*a);
    System.out.println("The eq has 2
    complex roots:" + real + "+" + imaginary
    + "i and" + real + "-" + imaginary + "i");
}
}
}
```

Date __/__/__

Page _____

```
class Main
{
public static void main(String[] args)
  {
    Quad q = new Quad();
    q.input();
    q.discriminant();
    q.calculateRoots();
  }
}
```

Output:

Quadratic equation is in the form : $ax^2 + bx + c$

Enter a : 2

Enter b : 6

Enter c : 3

~~No real soln, imaginary~~

~~Not of two 2 complex roots~~

Roots are real and unequal

First root is : -0.6339 7459

Second root is : -2.3660 2437

12/01/24

LAB - 2

```
import java.util.Scanner
class BOOK
{
    string name;
    String author;
    double price;
    int num_pages;

BOOK ( String name, String author, double price, int
        num_pages)
{
    this.name = name;
    this.author = author;
    this.price = price;
    this.num_pages = num_pages;
}

  void setDetails()
  {
     name = name;
     author = author;
     price = price;
     num_pages = num_pages;
  }

  void getDetails()
  {
     Scanner s = new
```

```
        Scanner (System.in);

        System.out.print("Enter Book Name:");
        name = s.nextLine();
        System.out.print("Enter Price:");
        price = s.nextDouble();
        System.out.print("Enter Number of Pages:");
        num_pages = s.nextInt();

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

    }

    public String toString()
    {
        return("Book Name:" + name + "\n
                Authors Name:" + author + "\n
                Price:" + price + "\n. Number of
                pages:" + num_pages);
    }
}

class BookMain
{
    public static void main(String args[])
    {
        Scanner s = new
        Scanner (System.in);
        int n, i;
        System.out.println("Enter Number of Books");
        n = s.nextInt();
        Book[] books = new Book[n];
```

Date __/__/__

Page _____

```
for(i=0; i<n; i++)
{
    System.out.println("Enter details of book."+(i+1));
    books[i] = new Book("","",0.0,0);
    books[i].getDetails();
}

for(i=0; i<n; i++)
{
    System.out.println("Details of book"+(i+1));
    System.out.println(books[i]);
}
}
}
```

Output

```
Enter number of books: 1

Enter name of book: Java
Enter name of Author: Tanmaji
Enter price: 999
enter Number of pages: 5

Book 1
Book Name: Java
Enter Author Name: Tanmaji
    Price: 999
Number of Pages: 5
```

Date __/__/__
Page

Student Details

```
import java.util.Scanner;
class Student
 {
    String USN;
    String name;
    int[] marks = new int[6];

 void acceptDetails()
  {
    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter USN: ");
      USN = scanner.next();

    System.out.print("Enter Name: ");
      name = scanner.next();

    System.out.println("Enter marks for 6 subjects");

  for(int i=0; i<6; i++)
  {
    System.out.print("Subject" + (i+1) + ": ");
      marks[i] = scanner.nextInt();
  }
}

double calculatePercentage()
 {
     int totalMarks = 0;
```

```
        for (int mark : marks)
        {
            totalMarks += mark;
        }

        return (double) totalMarks / 6;

    }

    void displayDetails()
    { System.out.println("Student Details");
      System.out.println("USN: " + USN);
      System.out.println("Name: " + Name);
      System.out.println("Percentage: " + calculatePercentage() + "%");
    }
}

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

        System.out.print("Enter the number of students: ");

        int numStudents = scanner.nextInt();

        Student[] students = new Student[numStudents];
```

```
for(int i=0; i<numStudents; i++)
{
    students[i] = new Student();
    System.out.println("\n Enter details for
    student" + (i+1) + ":");
    students[i].acceptDetails();
}

System.out.println("\n Details of all students:");
for(Student student : students)
{
    student.displayDetails();
    System.out.println();
}
}
```

Output

```
Enter number of students : 1
Enter USN : 1BM22CS001
Enter Name : Tanmayi
Enter marks for 6 subjects :
90 20 30 40 20 60

Student Details
USN: 1BM22CS001
Name: Tanmayi
Percentage : 76.54 %
```

LAB PROGRAM

Question → Create shape - asstract class
→ Three ceassis : Rectangle
Circle
Triangle
→ only printArea method contained

```
asstract class shape {
        private int length;
        private int breadth;
        shape (int length, int breadth)
        {
            this.length = length;
            this.breadth = breadth;
        }

asstract pubsic void printArea();
}

Class Rectangle entends shape {
    Rectangle (int length, int breadth)
    {
        super(length, breadth);
    }
pubsic void printArea() {
    System.Out.println (length * breadth);
}
}
```

Date __/__/__

Page _____

```
class Triangle extends shape
{
    Triangle ( int base, int height)
    {
        super (base, height);
    }
    public void printArea() {
        System.out.println (0.5* breadth * length);
    }
}

class circle extends shape { circle ( int radius ) {
        super(radius, 0);
    }

    public void printArea()
    {
        System.out.println (3.14 * length * length);
    }

public class main {
    public static void main ( String args [])
    {
        Rectangle r = new Rectangle (5,5);
        Triangle t = new Triangle (3,4);
        circle c = new Circle (5);

        Shape s;
        s = r;
        s.printArea ();
```

```
        s = t;
        s = printArea();
        s = c;
        s.printArea();
    }
}
```

Output

```
20
6.0
50.24
```

Question 2

Write a program to create a class Bank that maintains two kinds of account for its customers, one called saving account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name and type of account. From this derive the classes cur-acct and sav-acct to make them more specific to their requirements:

a) Accept deposit from customer and update the balance.
b) Display the balance.
c) Compute the deposit interest.
d) Permit withdrawal and update the balance.

Date __/__/__

Page _____

check for minimum balance, impose penalty if necessary and update the balance

CODE:

```
import java.util.Scanner;

class Account{
    protected string customerName;
    protected long accountNumber;
    protected string accountType;
    protected double balance;

public Account (string customername, long account
Number, string accountType, double balance)
{
    this.customerName = customerName;
    this.accountNumber = accountNumber;
    this.accountType = accountType;
    this.balance = balance;

public void displayBalance()
{
    System.out.println("Account Number: " +
                        accountNumber);
    System.out.println("Name: "+ customerName);
    System.out.println("Acc Type: " + accountType);
    System.out.println("Balance:"+ Balance);

}
```

```
public void deposit (double amount)
{
    balance += amount;
    System.out.println("deposit of amount $" +
                       + amount + " successful");

    displayBalance();

}
public void withdraw (double amount)
{
    if (amount <= balance)
    {
        balance -= amount;
        System.out.println("Withdrawal of
        amount $" + amount + "successfully");

    }

    else
    {
        System.out.println("insufficient funds");
    }
    displayBalance();

}

}

class CurrAct extends Account
{
    private double minimumBalance = 1000;
    private double serviceCharge = 50;
```

Date __/__/__
Page _____

```
public CurrAct(String customerName, long
        accountNumber, double balance)
{
    super(customerName, accountNumber,
            "current", balance);

}

public void withdraw(double amount)
{
    if(amount <= balance - minimumbalance)
    {
        balance -= amount;
    System.out.println("withdrawal of $" +
            amount + "successful");

}

else
 {
    System.out.println("insufficient funds.
        Service charge of $" + servicecharge +
            "imposed");

 }
displayBalance();

}
```

Date __/__/__

Page _____

```
class SavAcct extends Account
{
    private double interestRate = 0.05;
    public SavAcct ( String customerName, long
        accountNumber, double Balance )
    {
        super ( customerName, accountNumber, "Savings
            , balance );
    }

    void computeInterest ()
    {   double interest = balance.interestRate;
        balance += interest;
        displayBalance();
    }
}

pubeic class program1
{
    public static void main ( String[] args )
{   Scanner sc = new Scanner ( System.in );
    SavAcct SavingAct = new SavAcct ("John Doe", 12345, 500);

        SavingAct.displayBalance();
        SavingAct.deposit(1000);
        SavingAct.computeInterest();
        savingAct.withdraw(2000);

    CurAcct currentAct = new CurAcct ("Jane Doe", 98765,
                                      1500);

        currentAct.displayBalance();
        currentAct.deposit(500);
        currentAct.withdraw(2000);
```

```
sc.close()
    }
}
```

Output

```
Account Number : 12345
Customer Name : John Doe
Acc Type : Savings
Balance : $6000.0
Deposit of $1000 credited
Account Number : 123456
Customer Name : John Doe
Interest of $3000.0 credited
Account number : 987654
Customer Name : Jane Doe
Account Type : Current
Balance : $2000.0
* Insufficient funds.
Account Number : 987654
Customer Name : Jane Doe
Balance : $1950.0
```

Date
Page

LAB 16/02/2024

Question Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called 'Father' and a derived class called 'Son' which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge() when the input age < 0. In Son class, implement a constructor that takes both father and son's age and throws an exception if son's age >= father's age.

CODE:

```
// class for wrong age

class WrongAge extends Exception
{
    WrongAge() {
        super("invalid age provided");
    }
}

// Base class father
class Father {
    private int age;
    // constructor with exception handling for age
    Father(int age) throws WrongAge {
        if (age < 0) {
            throw new WrongAge();
        }
        this.age = age;
    }
    int getAge() {
        return age;
    }
}
```

Date __/__/__

Page _____

```
// derived class son extending Father
class Son extends Father {
        private int sonAge;

    // constructor with exception handling for sons age
    Son(int fatherAge, int sonAge) throws WrongAge {
        super(fatherAge);
        if (sonAge >= fatherAge)
        {
            throw new WrongAge();
        }
        this.sonAge = sonAge;
    }
    int getSonAge() {
            return sonAge;
    }
}

public class Main {
    public static void main(String[] args) {
        try {
            // creating a father instance with age 50
            Father father = new Father(50);
            System.out.println("Fathers age: " +
                                father.getAge());

        // creating a son instance with fathers age 50 and
        sons age 25
        Son son = new Son(father.getAge(), 25);
        System.out.println("sons age: " + son.getSonAge());
    } catch (WrongAge e) {
        System.out.println(e.getMessage());
    }
}
}
```

Date / / /
Page

Question

→ package CIE, SEE
→ CIE : class student, class internals (derived)
→ SEE : Externals (derived)

```
Package CIE;

Public Class Student
{
    string USN;
    string name;
    int sem;
}
Package CIE;
   Public class Internals extends Student
   {
       int[] internal marks = new int[5];
   }

Package SEE;

 import CIE.Student;

public class external extends student
{
    int[] external marks = new int[5];

}
```

Date __/__/__
Page _____

```
import CIE.internals
import SEE.Externals
Public class main
{

Public static void main (strings arg[])
{
                    Student
Internals.USN = "1BM22IC063";

Internals Student.name = "Tanmaji";

Internals Student.sem = 5;

                                    int
Internal student.internalMarks = new [] 
        { 80, 75, 95, 80, 75};

External externalstudent = new External();

externalstudent.USN = "1BM22IC443";

external student.name = "Jane";

external student.sem = 5;

external student.externalMarks = new int[]
        { 70, 80, 90, 90, 85};

}

}
```

Date / / /
Page

Question

Write a program which creates two threads, one thread displaying "BMS college of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.

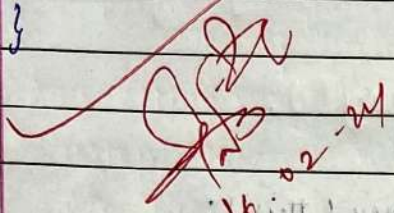
CODE:

```
class DisplayThread extends Thread {
    private String message;
    private int interval; // interval in milliseconds

    public DisplayThreads(String message, int interval)
    {
        this.message = message;
        this.interval = interval;
    }

    public void run() {
        while (true) {
            try {
                System.out.println(message);
                Thread.sleep(interval);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

public class Main {
    public static void main(String[] args) {
        DisplayThread thread1 = new DisplayThread("BMS
        college of Engineering", 1000);
```

Date __/__/__

Page _____

```
DisplayThread thread2 = new DisplayThread("CSE", 2000);

    thread1.start();
    thread2.start();
}
}
```



Output

BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE
CSE
CSE
CSE
CSE
CSE

LAB PRORAM

Program 9a

```
import java.awt.*;
import java.awt.event.*;
public class AWTExample extends WindowAdapter
{
    Frame f;
    AWTExample()
    {
        f.addWindowListener(this);
        Label l= new Label("employee id:");
        Button b= new Button("submit");
        TextFeild t= new TextFeild();
        l.setBounds(20,80,80,30);
        t.setBounds(20,100,80,30);
        b.setBounds(100,100,80,30);
        F.add(b);
        F.add(l);
        F.add(t);
        F.setsize(400,300);
        F.setTitle("employee info");
        F.setlayout(null);
        F.setVariable(true);
    }

public void windowclosing(windowEvent e)
{
        system.exit(0);

}
```

Date __/__/__

Page _____

```
Public static void main (String[] args)
{
    AWTExample awt_obj = new AWTExample();
}
}
```

Program 9b:

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

Public class EventHandling extends WindowAdapter
implements ActionListener
{
    Frame F;
    TextFeild tf;
    Event Handling()
    {
        F. = new Frame();
        F.addWindowListener(this);
        tF = new TestFeild();
        tF = setBounds(60,50,170,20);
        Button b = new Button("Click Me");
        b.setBounds(100,120,80,30);
        b.addActionListener(this);
        F.add(b);
        F.add(tb);
        F.setSize(300,300);
        F.setLayout(null);
        F.setVisible(true);
}
```

```
Public void windowClosing (WindowEvent e)
{
    System.exit(0);
}

Public static void main (String args[])
{
    new EventHandling();
}
```

Program 9c

```
import java.io.*;
Public class ByteArrayInput
{
    Public static void main (String[] args)
        throws IOException
    {
        byte[] buf = {35, 36, 37, 38};
        ByteArrayInputStream byt = new
                ByteArrayInputStream (byt);
        int k = 0;
        while ((k = byt.read()) != -1)
        {
            char ch = (char) k;
            System.out.println("ASCII value of
            character: " + k + "special
            character is: " + ch);
        }
    }
}
```

Date __/__/__

Page _____

Program 9d:

```
import java.io.*;
public class ByteArrayInput
{
    Public static void main (String[] args) throws
                                        IOException
    {
        byte[] buf = {35, 36, 37, 38};
    ByteArrayInputStream byt = new ByteArrayInput
                                    Stream (buf);
        int k = 0;
        while ((k = byt.read()) != -1)
        {
            char ch = (char)k;
        System.out.println("ASCII value of
    characters is" +k + " i special character is:" +ch);
        }
    }
}
```