1) 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.

12/12/2023

```
class overload{
    void print(int n)
    {
        int sum=0;
        for(int i=0; i<n; i++)
        {
            sum=sum+i;
        }
        System.out.println("sum of natural numbers is:"+sum);
    }

    void print(int m, int n)
    {
        System.out.println("Prime numbers 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 (20);
        O. print (4, 20);
    }
}
```

Output:

Sum of natural numbers is : 190

Prime numbers in the range are:

5
7
11
13
17
19

2a) write a java program to create a class grocery that has the variable c_name and c_phone. Create a method to accept 3 parameters to specify quantity of dal, quantity of pulses and quantity of sugar. The method to return the total price. Display the name, phno and total bill of 3 customers

```
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_amount (double a, double b, double c)
    {
        total = a*150 + b*80 + c*50;
    }
```

```
void display()
{
    System.out.println("Name"  ""+ "Phone no    "+Total");
    System.out.println(c_name+"     "+c_ph+".    "+total);
  }
}
```

Class Run1

```
{
  public static void main(String[] args)
  {
     Grocery g1 = new Grocery("rama", "7545426180");
     Grocery g2 = new Grocery("soma", "9874261264");
      g1.calc_amount(5,3,2);
      g1.display();
      g2.calc_amount(6,2,4);
      g2.display();
  }
}
```

Output

Name	Phone no	Total
rama	7545426180	1090.0

Name	Phone no	Total
Soma	9874261264	1260.0

3. Write a java program to calculate roots of a quadratic equation. Use appropriate methods to take input, and calculate the roots.

~~class quad~~

```
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.print("Enter a:");
     a=s.nextInt();
    System.out.print("Enter b:");
     b=s.nextInt();
     System.out.print("Enter c");
     c=s.nextInt();
      }
      void discriminant()
       {
           d=(b*b)-(4*a*c);
       }
      void calculateRoot(){
          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 :"+root1);
              System.out.println("Second root :"+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 solutions. Roots are imaginary");
double real = -b/(2*a);
double imaginary = Math.sqrt(-d)/(2*a);
System.out.println ("the equation has two complex roots:" +
real + " + " + imaginary + "i" and + real + " - " + imaginary + "i");
}
}
}
class main{
public static void main (String [] args)
{
Quad q = new Quad();
q.input();
q.discriminant();
q.calculateRoots();
}
}
```

output

first root is 2.0

second root is -4.0

a=6
b=12
c=-48

9. Create a class Book that contains four members: name, author, price, and numpages. Include a constructor to set the values for the members. Include methods to set and get details of object. Include a toString() method that could display the complete details of the book. Develop a Java program to create n book objects.

```
import java.util.Scanner;
Class Books
{
    String name, author;
    int price, numpages;

Books()
{ }

Books (String name, String author, int price, int numpages)
{
    this.name = name;
    this.author = author;
    this.price = price;
    this.numpages = numpages;
}
public String toString()
{
    String name, author, price, numpages;
    name = "book name:" + this.name + "\n";
    author = "Author name :" + this.author + "\n";
    Price = "Price:" + this.price + "\n";
```

```
numpages1 = "numpages:" + this.numpages + "\n";

return name + author + price + numpages1;
    }
}

class Book run{

Public Static void main (String args[])
{
    Scanner s = new Scanner (System.in);
    int n;

    String name, author;
    int price, numpages;

    System.out.println ("\n enter the number of books\n");
    n = s.nextInt();

    Books b[];
    b = new Books[n];

    for (int i=0; i<n; i++){
        Syste.out.println ("Book" + (i+1) + ":");
        System.out.println ("Enter name of book: ");
        name = s.next();
        System.out.println("Enter author:");
        author = s.next();
        System.out.println ("Enter price:");
        price = s.nextInt();
        System.out.println("Enter noof pages:");
        numpages = s.nextInt();
        b[i] = new Books(name, author, price, numpages);
    }
```

```
for(int i=0;i<n; i++)
    {
        System.out.println("Book"+(i+1)+":\n"+b[i]);
    }
  }
}
```

output

```
enter no of books :
1
Book 1 :
Enter name of book :
aaa
Enter author :
sss
Enter price :
1200
Enter no of pages :
140
Book 1 :
book name : aaa
Author name : sss
price : 1200
num of page : 140
```

1. write a java program to create a class student with members USN, name, marks[6 subjects]. Include methods to accept student details and marks, Also include a method to calculate the percentage and display appropriate details. (Array of student object

To be created.

```
import java.util.Scanner;

class Student
{
    String usn, name;
    int[] marks = new int[6];

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

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

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

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

        for(int i=0; i<6; i++)
        {
            System.out.println("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());
    }
}

Class studentExam{

    Public static void main (String[] args){

        Scanner scanner = new Scanner (System.in)

        System.out.println("Enter number of student:")
        int numstudents = scanner.nextInt();

        Student[] students = new student(numstudents);

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

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

output:

Enter no of students : 1

Enter details for student 1:

Enter USN : 110

Enter name : naa

Enter marks for 6 subjects:

S1 = 23

12 = 32

S3 = ~~100~~ 12

S4 = 35

S5 = 32

S6 = 65

Details of students

USN : 110

name: naa

percentage : 37.16666%

1. Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and circle such that each one of the classes extends the class Shape. Each one of the classes contain the method printArea() that prints the area of the given shape.

```
abstract class Shape
{
    int side1,side2;

    Shape(int s1, int s2);
    {
        side1=s1;
        side2=s2;
    }

    abstract void printArea();
}

class Rectangle extends Shape{
    Rectangle (int length, int width)
    {
        super (length, width);
    }

    void printArea()
    {
        int area = side1*side2;
        System.out.println("Area of Rectangle:"+area);
    }
}
```

```
class Triangle extends Shape
{
    Triangle (int base, int height)
    {
        super (base, height); }

    void printArea()
    {
        double area = 0.5 * side1 * side2;
        System.out.println ("Area of Triangle:" +area);
    }
}

class Circle extends Shape
{
    Circle (int radius)
    { super (radius, 0);
    }

  void printArea()
    {
      double Area = Math.PI * side1 * side1;
      System.out.println ("Area of circle" +area);
    }
  }

public class ABSrum
{
  public static void main (String[] args)
    {
      Rectangle rectangle = new Rectangle (5,8);
      Triangle triangle = new Triangle (4,6);
      Circle circle = new Circle (3);

      rectangle.printArea();
      triangle.printArea();
      Circle.printArea();
    }
}
```

Output:

Area of Rectangle : 40

Area of Triangle : 12.0

Area of Circle : 28.274

2. Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides check book facilities 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, account number and type of account. From this derive the classes cur-acct and sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

a) Accept deposit from customer and update the balance

b) Display the balance.

c) Compute and deposit interest

d) Permit withdrawal and update the balance.

Check for the minimum balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;

class Account{

    String customerName, accountType;
    Long accountNumber;
    double balance;

    Account(String customerName, long aN, String aT, double b)
    {
        this.customerName = customerName;
        accountNumber = aN;
        accountType = aT;
        balance = b;
    }

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

    void deposit(double amount):
    {
        double t = amount;
        System.out.println("Deposit of:" + amount + "success");
        displayBalance();
    }
    void withdraw(double amount)
    {
        if(amount <= balance)
        {
            balance = balance - amount;
            System.out.println("Withdrawal of" + amount +
                               "successful");
            displayBalance();
        }
```

```
        else {
            System.out.println("Insufficient fun de. withdrawal fail");
        }
        display Balance();
    }
}

class CurrAcct extends Account {
    double minimumBalance = 1000, @ ServiceCharge = 50;

    CurrAcct (String cN, long aN, double b)
    {
        super(cN, aN, "current", b);
    }

    void withdraw (double amount)
    {
        if (amount <= balance - minimBlance)
        {
            balance -= amount;
            System.out.println ("withdrawal of" +amount+" successful
        }
        else {
            System.out.println ("Insufficient funds, Service charge of
                                Service charge + "imposed");
            balance -= serviceCharge;
        }
        displayBalance();
    }
}

class SaveAcct extends Account {
    double interestrate = 0.05;

    SaveAcct (String cN, long aN, double b)
    {
        super (cN, aN, "Saving", b)
    }
```

```
void computeInterest()
{
    double interest = balance * intrestRate;
    balance += interest;
    System.out.println("Interest of "+interest+" credited");
    displayBalance();
}
}

class Bank
{
  public static void main(String[] args)
  {
    Scanner scanner = new Scanner(System.in);
    SavAcct sa = new SavAcct("hari", 16464123, 5000);
    sa.displayBalance();
    sa.deposit(1000);
    sa.computeInterest();
    sa.withdraw(3000);

    CurrAcct CA = new CurrAcct("shyam", 9894214098, 100);
    CA.displayBalance();
    CA.deposit(2000);
    CA.withdraw(1100);
  }
}
```

output:

Account Number: 1646461 23

Customer Name : hari

Account Type : Saving

Balance : 5000.0

Deposit of 1000.0 successful.

Balance : 6500

Deposit of 3000.0 successful.

Account Number : 98421402

Customer Name: Shyam

Account Type : Current

Balance : 35000

Account Number: 1646461 23

Customer Name: hari

Account Type : Savings

Balance : 8000.0

Interest of 3000 credited.

19.01.24

Account Number : 1646461 23

customer name : hari

Account type: saving

balance : 63000.

withdrawal of 3000.0 successful.

Account Number : 1646461 23

Customer Name : hari

Account type : Saving

balance : 3300.0

Account number : 98421402

customer name : Shyam

Account Type : current

balance : 1500.0

Deposit of 2000.0 successful.

Account Number : 98421402

Customer Name : Shyam

Account Type : current

16/2/2024

week-5

1) CIE - Package

```
Package CIE;

Public class Personal
{
    public string usn, name;
        int sem;

    Personal (String usn, String name, int sem)
    {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}

import java.util.Arrays;

Class internals
{
    int[] internal_marks;

    internals ( int[] a;)
    {
        internal_marks = ;;
    }
}

Package SEE;
import CIE.Person;
public class external extends Personal
{
    int[] seemarks;
    external (String usn, String name, int sem, int[] seemarks)
    {
        super (usn, name, sem);
        this.seemarks = seemarks;
    }
}
```

```
package Formain;
import CIE.*;
import SEE.External;

class Run
{
    public static void main(String[] args)
    {
        int n=3;
        Student[] students= new Student[n];
        for(int i=0;i<n;i++)
        {
            int[] internal_marks ={80,90,70,60,80};
            int[] secmarks = {70,80,75,90,85};
            students[i] = new Student(new Personal("usn"+i,
            "student"+i,1), new Internal(internalmarks));

            Student[i].see = new External("usn"+i,"student"+i
                                secmarks);
        }
        for(int i=0; i<student length; i++)
        {
            student student = students[i];
            sop("student"+ student.personalname);
            sop("internal marks:" + Arrays.toString(
                    internals.internalmarks));
            sop("see Marks:" Arrays.toString(student.[i
                = see.marks]);
        }
    }
```

```
static class Student
{
public Personal personal;
public Internals internals;
public External see;
public Student (Personal personal, Internals internal)
{
    this.personal = personal;
    this.internals = internal;
}
}
```

output:

Student : Student 0

Internal marks (80, 90, 70, 60, 80);

SEE marks (70, 80, 75, 90, 85);

Student : Student 1

Internal marks (80, 90, 70, 60, 80);

SEE marks (70, 80, 75, 90, 85);

Student : Student 2:

Internal marks (80, 90, 70, 60, 80);

SEE marks (70, 80, 75, 90, 85);

2) Write a program that Demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "son" which extends the base class. In Father class, implement Constructor which takes the age and throws an exception Wrong age !! when the input age<0. In son class, implement a constructor that cases both father and son's

age and throw an exception if son's age is >= father's age.

```
import java.util.Scanner;
class Father
{
    int age;
    Father (int age)
    {
    ~~this.age = age;~~
      if (age<=0)
      {
        throw new IllegalArgumentException("age cannot b
          0 or negative");
        }
        this.age = age;
      }
}

Class Son extends Father
{
   int sonage;
    Son (int fatherage, int sonAage);
      {
        super (fatherage);
        if (sonage >= FatherAge)
        {
          throw new IllegalArgumentException ("Sons age cannot
                                              be greeather than
                                              father's age")
        }
        this.Sonage = sonage;
      }
}

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

```
try{
    sop("Enter Father age:");
    int fage = s.nextInt();
    sop("Enter Son age:");
    int sage = s.nextInt();
    Son ss = new Son(fage, sage);
    sop("Father age" + son.age);
    sop("Son age" + son.son age);
}
catch(IllegalArgumentException e)
{
    sop("Except:" + e.getmessage());
}
}
}
```

o/p:

① Father's age: 40
Son's age: 15
Father's age: 40
Son's age: 15

② Father's age: 40
Son's age: 60
Son's age cannot be greater than father age;

5) Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying 'CS' once every two seconds.

```
class bmsce extends Thread
{
    public void run()
    {
        try{
```

```
for(int i=0; i<1; i++)
{
    System.out.println("Bmscollege");
    Thread.sleep(1000);
}
}
catch (InterruptedException e)
{
    System.out.println(e);
    Thread.sleep(2000);
}
}
}
class main
{
public static void main(String a[]);
{
    bmsce bm = new bmsce();
    cs c = new cs();
    bm.start();
    c.start();
}
}

class cl extends Thread
{
    public void run();
    {
    try{
    for(int i=0; i<5; i++)
    {
        System.out.println("CSE");
        Thread.sleep(2000);
    }
    }
    catch(InterruptedException e)
    {
        System.out.println(e);}}}
```

CSE

O/P: BMS college.

ISE

CSE

ISE

CSE

BMS college

BMS college

BMS college

BMS college.