Program-1

```
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
                                numbers is " + sum);
    }
    void print (int m, int n) {
        System.out.println(" Prime number in the
        for (int i=m; i<=n; i++) {          range are ");
            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);
    }
}
```

DOHS

Program-2

```
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 + sugar * 50;
}
void display(){
        System.out.println("Name" + "t" "Phone number" + " " + "
                                                    Total");
        System.out.println(C_name + "  " + C_ph + "  " + total);
        System.out.println();
    }
}
class GDare {
        Public static void main (String [] args){
        Grocery g1 = new Grocery ("Rana", "806307");
        Grocery g2 = new Grocery ("Sharma", "7863");
        Grocery g3 = new Grocery ("Bhavan", "923357");
         g1.calc (2,1,2);
         g1.display ();

         g2.calc (3,5,2);
         g2.display ();
    }
}
```

Program - 3

```
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.print("Enter b: ");
    b = s.nextInt();
    System.out.println("Enter c: ");
    c = s.nextInt()
  }
void discriment() {
        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);
```

DOMS

```
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 solution. 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 {
    PSVM(String args[]){
        Quad q = new Quad();
          q.input();
          q.discriminant();
          q.calculateRoots();
        }
    }
```

Program - 4

```
import java.util.Scanner;
class Books
{
  String name;
  String author;
  Int price;
  Int 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";
   numPages = "number of pages :" + this.numPages + "\n";
   return name + author + price + numPages;
  }
}
Class Main{
   PSVM (String args[]){
      Scanner S = new Scanner(System.in);
      int n;
```

```
String name;
String author;
int price, numPages;
S.o.p ("Enter the no. of books");
n = S.nextInt();
Books b[];
b = new Books[n];
for (int i=0; i<n; i++)
{
   S.o.p("Book" + (i+1) + ":");
   S.o.p("Enter name of book:");
   name = S.next();
   S.o.p("Enter author:");
   author = S.next();
   S.o.p("Enter price");
   price = S.nextInt();
   S.o.p("Enter no of pages");
   numPages = S.nextInt();
   b[i] = new Books(name, author, price, numPages);
}
for (int i=0; i<n; i++)
      S.o.p("Book" + (i+1) + ":\n" + b[i]);
}
}
```

Enter the no. of books : 2
Book 1:
Enter the number of book: Jungle_Book
Enter the author: Rudyard
Enter the price : 1000
Enter the number of pages : 500

Book 2:
Enter the name of the book: Akbar & Birbal
Enter the author : Birbal
Enter the price : 2000
Enter the number of pages: 5000

Book 1:
Book name: Jungle - Book
Author name: Rudyard
Price : 1000
Number of pages : 500

Book 2:
Book name: Akbar & Birbal
Author name : Birbal
Price : 2000
Number of pages : 5000

Program -5

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

    void input details()
    {
        Scanner scanner = new Scanner (System.in);
        System.out.println("Enter USN");
        USN = scanner.nextLine();
        System.out.println("Enter marks of 6 subjects");
        for (int i=0; i<6; i++)
        {
            System.out.println("Subject" + (i+1) + ":");
            marks[i] = Scanner.nextDouble();
        }
    }

    double calculate percentage()
    {
        double total marks=0;
        for (double marks : marks)
        {

            total Marks += marks;
        }
    return (total marks/6);
```

```
void display details ()
{
    System.out.println("Student Details");
    System.out.println("USN:" + USN);
    System.out.println("Name" + Name);
    System.out.println("Percentage" + calculate Percentage()
                                        + "%");
}
}
class OOT
{
  public static void main (String [] args)
    {
      Scanner scanner = new Scanner (System.in);
    System.out.println("Enter the number of students");
    int number of Students = scanner.nextInt();
    Student [] student = new Student [number of Students]
    for (int i=0; < number of students ; i++)
      {
        System.out.println("Enter details for Student
                                    " + (i+1) + ":");
        Student [i] = new Student ();
        student [i].input Details ();
      }
      for (Student istudent : students)
        {
          student display details ();
        }
    }
}
```

Output:-

Enter the number of students: 2

Enter details for student 1:
Enter the USN: 1BM22CS285
Enter name: ABC
Enter marks for 6 subjects:
Subject 1: 78
Subject 2: 87
Subject 3: 90
Subject 4: 99
Subject 5: 79
Subject 6: 80

Enter details for student 2:
Enter USN: 1BM22CS240
Enter name: XYZ
Enter marks of 6 subjects
Subject 1: 55
Subject 2: 67
Subject 3: 76
Subject 4: 80
Subject 5: 77
Subject 6: 76

Student details:
USN: 1BM22CS285
Name: ABC
Percentage: 85.5%

Student details
USN: 1BM22CS400
Name: XYZ
Percentage: 71.16666666666667

Program-6

1) Develop 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 contains the method print area() that prints the area of the given shape.

```
abstract class shape
{
    int length;
    int width;
    shape (int length, int width)
    {
        this.length = length;
        this.width = width;
    }
    abstract void printArea();
}
class rectangle extends shape
{
    rectangle (int length, int width)
    {
        super (length, width);
    }
    void printArea()
    {
        int area = length * width;
        System.out.println("Rectangle area:" + area);
    }
```

```
}
class triangle extends shape
{
  triangle(int length, int width)
  {
    super (length, width)
  }
    void print Area()
    {
      double area = 0.5* length * width;
      System.out.println("Triangle area:" + area);
    }
}
class circle extends shape
  {
      circle (int length)
      {
          super(length));
    }
    void print Area()
    {
      double area = Math.PI* length* length;
      System.out.println("circle area" +area);
    }
}
  class main
{
  public static void main(String args[])
    {
      Rectangle r = new Rectangle (5, 10);
        r.print Area();
```

```
Triangle t = new Triangle(4,6);
t.printArea();
Circle c = new Circle(3);
c.printArea()
    }
}
```

Output:-
Rectangle area: 50
Triangle area: 12
Circle area : 28.26

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 provide compound interest and withdrawal facilities but no cheque book facility. the current account provide cheque book faculty 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 current account and savings acc to make them more specific to their requirements. Include the necessary in order to achieve 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 balance

```
class Account
{
    String customerName;
    int accountNumber;
    String accountType;
    double balance;

    Account(String customerName, int accountNumber,
            String accountType, double balance)

    {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = balance;
    }
    void deposit(double amount)
    {
        balance += amount;
        System.out.println("Deposit successful. Updated
                            balance: " + balance);

    }
    void displayBalance()
    {
        System.out.println("Account Number:" + accountNumber);
        System.out.println("Balance:" + balance);
    }
}
class SavAcc extends Acc
{
    double interestRate;
```

```
SavAcct (String customerName, int accountNumber, double
         balance)
{
  super (customerName, accountNumber, "Savings", balance);
  this.interestRate = 0.05;

}
 void depositInterest()
{
  double interest = balance * interestRate;
  balance += interest;
  System.out.println("Interest deposited. Updated balance"
                     + balance);
  }
  void withdraw(double amount)
  {
     if (balance >= amount)
     {
        balance -= amount
        System.out.println("Withdrawal successful
                            Updated balance: " + balance);

      }
     else
     {
     System.out.println("Insufficient funds for
                         withdrawal.");

     }

  }
}
```

DOMS

Program-7

1) Write a program that demonstrates handling of exceptions of inheritance tree. Create a base class called "father" and derived class called "Son" which extends the base class. In father class, implement a constructor which takes the age and throws the exception wrong age() when the input age<0. In Son class implement a constructor that cases both father and son age and throws an exception if Sons age is >= fathers age.

```
class father {
public int age;
father (int age) {
  if (age>0) {
    throw new [illegal Argument Exception] ("Age wrong Age
                                            cannot be negative");
  }
  this.age = age;
}
}

[public] class son extends father {
  public int Son age;
  public son(int fatherage, int Son age) {
    super (father age);
    throw new [Illegal Argument Exception] ("son's age cannot
                                            be >=fathers age");
  }
  this son age = Son age;
}
}
```

```
import java.util.Scanner;
public class main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        try {
        System.out.println("Enter father age:");
        int fatherAge = scanner.nextInt();
        System.out.println("Enter son's age:");
        int sonAge = scanner.nextInt();

    Son son = new son(fatherAge, sonAge);
        System.out.println("Fathers age:" + son.age);
        System.out.println("Sons age:" + son.sonAge);
    }

    catch(IllegalArgumentExcption e) {
        System.out.println("Exception" + e.getMessage());
    }
    scanner.close();
    }
}
```

Output:

```
Enter fathers age: 20
Enter son's age: 40
Exception. son's age cannot be >= father's age
```

2) Write a program which creates two threads one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSE" once every 2 seconds.

```
class Threads extend thread {
  private String sent;
  private int intervals;
  threads (string sent, int intervals) {
    this.sent = sent;
    this.intervals = intervals
  }
  public void run() {
    try {
      while(true) {
          System.out.println(sent);
          Thread.sleep(1000);
        } catch (InterruptException e) {
            System.out.println(e);
        }
    }
  }
class main {
    public static void main (String [] args) {
      Threads thread1 = new Threads ("BMS college", 10);
      Threads thread2 = new Threads ("CSE", 2)
      thread1.start();
      thread2.start();
    }
}
```

Output :-

MMS college
CSE

BMS college
CSE

,

,

,

Packages:-

```
package CIE;
import java.util.Scanner;
public class Student {
protected String USN = new String ();
protected String name = new String ();
protected int sem;

public void inputStudentDetails ()
{
Scanner sc = new Scanner (System.in);
System.out.println(&amp;quot; Enter student
USN: sc.next();          USN &amp;quot;);
System.out.println (&amp;quot; Enter student name
                    &amp;quot;);
name= sc.next ();
System.out.println(&amp;quot; Enter student
                    Semester &amp;quot;);
Sem = sc.nextInt();
}
public void displayStudentDetails () {
System.out.println(&amp;quot; student USN : &amp;quot;+usn);
System.out.println(&amp;quot; student name: &amp;quot;+name);
System.out.println(&amp;quot; student sem: &amp;quot;+sem);
}
}
```

```
package CIE;
import java.util.Scanner;
public class internals extends Student {
protected int marks[] = new int[5];
public void input CIE marks ()
```