Date __/__/__
Page _____

1) Develope a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read in a, b, c & use the quadratic formula. If the discriminate b^2-4ac is negative, display a message stating that there are no real solutions.

```
import java.util.*;
class Quadratic
{
  public static void main (String arg[])
  {
    Scanner sc = new Scanner (System.in);
    System.out.println ("Enter the co-efficient of a:");
    double a = sc.nextDouble();
    System.out.println ("Enter the co-efficient of b: ");
    double b = sc.nextDouble();
    System.out.println ("Enter the co-efficient of c:");
    double c = sc.nextDouble();
    double d = b*b - 4*a*c;
    if (d>0)
    {   double r1 = (-b + Math.sqrt(d))/2*a;
        double r2 = (-b + -Math.sqrt(d))/2*a;
        System.out.println ("Roots are " +r1+ "and "+r2);
    }
    else if (d==0)
    { double r1 = -b/2*a;
      double r2 = -b/2*a;
      System.out.println("Roots are "+r1+"and "+r2);
    }
    else if (d<0)
    { System.out.println("There are no real solution");
    }
```

```
        else
        { System.out.println("INVALID INPUT");
        }
    }
}
```

Output:

```
Enter the co-effecient a : 1
Enter the co-efficient b : -3
Enter the co-efficient c : 2
The equation has roots 2.0 and 1.0
```

2) Develope a Java
with members
marks. Includ
a method to c

```
import java.uti
class Student
{
    Strin
    Strin
    int
    int
    int

    public
    { Scann
      Syste
      usn
      Sys
      nam
      Sy
      nu
      cre
      m
      for
      {
```

2) Develope a Java program to create a class Student with members usn, an array credits & an array marks. Include methods to accept & display details & a method to calculate SGPA of a student.

```
import java.util.Scanner;
class Student
{
    String usn;
    String name;
    int[] credits;
    int[] marks;
    int numsubs;

    public void acceptDetails()
    {   Scanner sc = new Scanner(System.in);
        System.out.println("Enter USN: ");
        usn = sc.nextLine();
        System.out.println("Enter name: ");
        name = sc.nextLine();
        System.out.println("Enter the number of subjects:");
        numsubs = sc.nextInt();
        credits = new int[numsubs];
        marks = new int[numsubs];
        for(int i=0; i<numsubs; i++)
        {
            System.out.println("Enter credits: ");
            credits[i] = sc.nextInt();
            System.out.println("Enter marks: ");
            marks[i] = sc.nextInt();
        }
    }
```

```
public void display()
{
    System.out.println("USN : " +usn);
    System.out.println("Name : " +name);
    System.out.println("No. of subjects : " +numsubs);

    for(int i=0; i<numsubs; i++)
    { System.out.println("Subject " +(i+1)+
                "Credits : " +credits[i]+ "Marks:"
                +marks[i]);
    }
}

public double SGPA()
{  int totalcred = 0;
   int totalgradepts = 0;

   for(int i=0; i<=numsubs ; i++)
   {  int gradepoint = getgradepoint marks[i];
      totalgradepts += gradepoint + Credit[i];
      totalcred += credits[i];
   }
   if(totalcred>0)
   { return double Totalgradepts/Totalcredits;
   }
   else
   { return 0.0;
   }
}

public int getgradepoint(int marks)
{  if(marks >=90)
      return 10;
```

Date __/__/__
Page _____

```
else if (marks >= 80)
    return 9;
else if (marks >= 70)
    return 8;
else if (marks >= 60)
    return 7;
else if (marks >= 50)
    return 6;
else if (marks >= 40)
    return 5;
else
    return 0;
}
}
public class Main
{ public static void main (String args[])
  { Student s = new Student();
    s.acceptDetails();
    s.display();
    double sgpa = s.SGPA()
    System.out.println("SGPA : " + sgpa);
  }
}
```

Date
Page

Output:

Enter USN: 1BM23ET045
Enter name: Saahya K S
Enter number of subjects: 3
Enter credits for subject 1: 4
Enter marks for subject 1: 85
Enter credits for subject 2: 3
Enter marks for subject 2: 75
Enter credits for subject 3: 3
Enter marks for subject 3: 66

Student Details:
USN: 1BM23ET045
Name: Saahya K S
Number of subjects: 3
Subject 1: credits = 4, Marks = 85
Subject 2: credits = 3, Marks = 75
Subject 3: credits = 3, Marks = 66
SGPA: 8.1

Date __/__/__
Page _____

3) Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set & get the details of the objects. 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 Book
{
    String name;
    String author;
    double price;
    int num_pages;

    Book(String n, String a, double pr, int pg)
    {   name = n;
        author = a;
        price = pr;
        num_pages = pg;
    }

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

Date / /
Page

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

  for (int i=0; i<n; i++)
  { System.out.println("Enter Name : ");
    String name = sc.nextLine();
    System.out.println("Enter Author : ");
    String author = sc.nextLine();
    System.out.println("Enter the price : ");
    Double price = sc.nextDouble();
    System.out.println("Enter the Number of
      pages : ");
    int num-pages = sc.nextInt();

  books[i] = new Book(name, author, price,
                num-pages);
  }

  for (int i=0; i<n; i++)
  {System.out.println(books[i].toString());
  }
}
}
```

Date __/__/__
Page _____

Output:

Enter number of books : 2
Enter details for Book 1
Enter name : A Good Girl's Guide To Murder.
Enter Author : Holly Jackson
Enter Price : 499 rupees.
Enter number of pages : 365

Enter details for Book 2
Enter name : Five Survive.
Enter Author : Holly Jackson.
Enter Price : 300 rupees
Enter number of pages : 250

-- Book List --

Book 1
Book Details :
Name : A Good Girl's Guide To Murder.
Author : Holly Jackson
Price : 499 rupees
Number of pages : 365

Book 2
Book Details :
Name : Five Survive
Author : Holly Jackson
Price : 300 rupees
Number of pages : 250.

Date __/__/__
Page

4) 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 ~~printArea()~~ Rectangle, Triangle & Circle such that each one of the classes contain only the method printArea() that prints the area of the given shape.

```
import java.util.Scanner;
abstract class Shape {
    int dim1;
    int dim2;
    abstract void ~~printf(creat);~~ printArea();
}
class Rectangle extends Shape
{   Rectangle (int length, int breadth)
    {   this.dim1 = length;
        this.dim2 = breadth;
    }
    void ~~printf(creat())~~ printArea()
    {   int area = dim1 * dim2;
        System.out.println("Area of the Rectangle :"
                                    + area);
    }
}

class Triangle extends Shape
{   Triangle (int base, int height)
    {   this.dim1 = base;
        this.dim2 = height;
    }
```

```
    void printArea()
    { double area = 0.5 * dim1 * dim2;
      System.out.println("Area of triangle: "
                         + area);
    }
}
class Circle extends Shape
{   Circle(int radius)
    { this.dim1 = radius;
    }
    void printArea()
    { double area = 3.14 * dim1 * dim1;
      System.out.println("Area of the circle: " +
                         area);
    }
}
public class Main
{   public static void main(String args[])
    { Scanner sc = new Scanner(System.in);
      System.out.println("Enter length & breadth of the
                         Rectangle: ");
      int l = sc.nextInt();
      int b = sc.nextInt();

      Rectangle rect = new Rectangle(l, b);
      rect.printArea();
      System.out.println("Enter base & height of the
                         Triangle: ");
      int ba = sc.nextInt();
      int h = sc.nextInt();
      Triangle tr = new Triangle(ba, h);
      tr.printArea();
```

```
        System.out.println("Enter radius of the circle:");
        int r = sc.nextInt();
        Circle c = new Circle(r);
        c.printArea();
    }
}
```

Output:

Enter length of Rectangle: 5
Enter breadth of Rectangle: 4
Area of Rectangle = 20

Enter base of Triangle: 6
Enter height of triangle = 3
Area of Triangle = 9

Enter radius of circle: 7
Area of Circle: 153.93800000000002

Date __/__/__
Page _____

5) Develop a Java program to create a class Bank that maintains two kinds of account for its 10 customers, one called savings account & the other account called current account. The savings account provides compound interest & withdrawal facilities but no cheque book facilities. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance & if the balance falls below this level, a service charge is imposed.

Create a class Account that stores customer name, account number & type of account. From this derive the classes cur-acct & 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 & update the balance.
(b) Display the balance.
(c) Compute & deposit interest.
(d) Permit withdrawal & update the balance.

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

```
import java.util.Scanner;
class Account
{ String customerName;
  String AccNo;
  String acctype;
  double balance;
  Account(String cname, String anum; String
          atype)
  { customerName = cname;
    AccNo = anum;
    acctype = atype;
```

Date __/__/__
Page _____

5) Develop a Java program to create a class Bank that maintains two kinds of account for its to customers, one called savings account & the other account called current account. The savings account provides compound interest & withdrawal facilities but no cheque book facilities. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance & if the balance falls below this level, a service charge is imposed.

Create a class Account that stores customer name, account number & type of account. From this derive the classes cur-acct & 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 & update the balance
(b) Display the balance.
(c) Compute & deposit interest.
(d) Permit withdrawal & update the balance.
Check for the minimum balance, Impose penalty if necessary & update the balance.

```
import java.util.Scanner;
class Account
{ String customerName;
  String AaccNo;
  String acctype;
  double balance;
  Account(String cname, String anum; String
          atype)
  { customerName = cname;
    accNo = anum;
    acctype = atype;
```

```
        balance = 0.0;
    }
    public void deposit (double amount)
    { if (amount >0)
      { balance += amount;
        System.out.println ("Deposited: " + amount);
      }
      else
      { System.out.println ("Invalid amount ");
      }
    }
    public void displayBalance ()
    { System.out.println ("Balance :  " + balance);
    }
    public void updateBalance (double amount)
    {  balance = amount;
    }
}

class SavAcct extends Account
{    double interestrate;
    public SavAcct (String customerName, String
                    accNo, double interest)
    { Super (customerName, accNo, "savings");
      this.interestrate = interest;
    }
    public void computeInterest (int years)
    { double balance = getBalance();
      double interest = balance * Math.pow((1+interestrate(
                                    years) - balance;
      deposit (interest);
      System.out.println ("Interest of rupees " +
                          interest + " added ");
    }
```

Date __/__/__
Page _____

```
public void withdraw (double amount)
 { double balance = getBalance();
   if (amount > 0 && amount <= balance)
   { updateBalance (balance - amount);
     System.out.println("Withdrawn : " + amount);
   }

  else
   { System.out.println ("Insufficient balance");
   }

 }
}


class CurrAcct extends Account
{  double minBalance;
   double penalty;

   CurrAcct (String customerName, String accNumber,
             double minBalance, double penalty)
  { super (CustomerName, accountNumber, "current");
    this.minBalance = minBalance;
    this.penalty = penalty;
  }

  public void withdraw (double account)
   { double balance = getBalance();
     if (amount > 0 && amount <= balance)
     { updateBalance (balance - amount);
       System.out.println (" Withdrawn :  " + amount);
       if (getBalance() < minBalance)
       { updateBalance (getBalance() - penalty);
         System.out.println ("Balance below minimum
                              penalty of " + penalty);
       }
```

100),

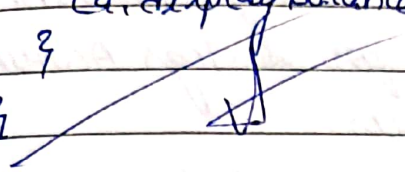
```
else
{ System.out.println("Insufficient balance "); // or
                                    invalid amount ");
}

public class Main
{  public static void main(String args[])
   { Scanner sc =new Scanner(System.in);
     System.out.println("Enter name for savings
                                    account");
     String savName = sc.nextLine();
     System.out.println("Enter the amount for
                                    Savings account = ");
     String savAccNum = sc.nextLine();
     SavAcct sa = new SavAcct(savName, savAccNum,
                                    9.0);
     sa.deposit(1000);
     sa.computeInterest(2);
     sa.displayBalance();
     sa.withdraw(500);
     sa.displayBalance();
     System.out.println("Enter customer name for
                                    Current account : ");
     String curName = sc.NextLine();
     System.out.println("Enter account numbers for
                                    current account: ");
     String currAccNum = sc.nextLine();

     CurrAcct ca = new CurrAcct(curName, currAccNu
                                    500, 50);
     ca.deposit(1500);
     ca.displayBalance();
     ca.withdraw(1200);
     ca.displayBalance();
```

Date __/__/__
Page _____

```
ca.withdraw(500);
ca.displayBalance();
}
}
```



Date / /
Page

Output:

Enter customer name for Savings Account: Anne
Enter account number for Savings Account: 5123
Deposit of = 1000.0 Rupees
Interest of Rupees 81.599 added to your account
Balance : 1081.6 Rupees
Withdrawn : 500.0 Rupees
Balance : 581.6 Rupees.

Enter customer name for Current Account : Noah
Enter account number for Current Account : 7862
Deposited : 1500.0 Rupees
Balance : 1500.0 Rupees
Withdrawn : 1200.0 Rupees
Balance : 300.0 Rupees
Balance below minimum Penalty of 50.0 Rupees Imposed
Balance : 250.0 Rupees.
Withdrawn : 500.0 Rupees
Insufficient balance or Invalid amount
Balance : 250.0 Rupees

28/11/24

Date __/__/__
Page

6) Create a package CIE which has two classes - Student & Internals. The class Internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

```
package CIE;
public class Student
{   public String usn;
    public String name;
    public int sem;

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

public void displayStudentDetails ()
{ System.out.println("USN: " +usn);
  System.out.println("Name: " +name);
  System.out.println("Sem: " +sem);
}
}

public class Internals
{ public int[] InternalMarks = new int[5]
```

```
public void displaySEEMarks()
{ System.out.print("SEE Marks: ");
  for(int i=0; i < seeMarks.length(); i++)
  { System.out.print(Marks[i] + " ");
  }
  System.out.println();
}
}
```

—x—

```
import CIE.*;
import SEE.*;
import java.util.Scanner;

public class Main {
  public static void main(String args[])
{   Scanner sc = new Scanner(System.in);
    System.out.println("Enter number of
                        students: ");
    int n = sc.nextInt();
External[] Students = new External[n];
Internals[] internals = new Internals[n];

for(int i=0; i<n; i++)
{ System.out.println("\n Enter details of
                      student: ");
  System.out.println("Enter USN: ");
  String usn = sc.next();
  System.out.println("Enter Name: ");
  String name = sc.nextLine();
  System.out.println("Enter Sem: ");
  int sem = sc.nextInt();
```

```
Students[i] = new E
internals[i] = new

System.out.println(
int[] internalMa
for(int j=0; j<
  { internalMark
  }
internals[i].set

System.out.print
int[] seeMarks =
for(int j=0; j
  { seeMarks(
  }
  student[i].

System.out.pri
for(int i=0
  { System.ou
  students[i
  Internals[
  student[i

System.out

for(int j
  { int fi

  System
  }
  System.
}
```

Date __/__/__
Page _____

```
Students[i] = new External(usn, name, sem);
Internals[i] = new Internals[];

System.out.println("Enter 5 internal marks:");
int[] internalMarks = new int[5];
for(int j=0; j<5; j++)
  { internalMarks[j] = sc.nextInt();
  }
internals[i].setInternalMarks(internalMarks)

System.out.println("Enter 5 SEE Marks:");
int[] seeMarks = new Int[5];
for(int j=0; j<5; j++)
  { seeMarks(j) = sc.nextInt();
  }
  Student[i].setSEEMarks(seeMarks);

System.out.println("\n Final Marks of Student");
  for(int i=0; i<n; i++)
  { System.out.println("\n Student"+(i+1)+":");
    students[i].displayStudentDetails();
    Internals[i].displayInternalMarks();
    student[i].displaySEEMarks();

    System.out.print("Final Marks:");

    for(int j=0; j<5; j++)
    { int finalMarks = internals[i].internalMarks
                      +(students[i].seeMarks[j]/2);  [j]{
      System.out.print("finalMarks+" ");
    }
    System.out.println();
  }
}
```

Output:

Enter number of students : 1
Enter USN : 1BM23ETD45
Enter Name : Sadhya K S
Enter Sem : 3
Enter 5 internal marks :
20 25 18 23 20
Enter 5 SEE marks :
50 55 40 45 35

Student 1 :
USN : 1BM23ETD45
Name : Sadhya K S
Semester : 3
InternalMarks : 20, 25, 18, 23, 20
SEE Marks : 50, 55, 40, 45, 35
Final Marks : 45, 52, 38, 45, 37.

7) Write a
exception
class ca
"son" whi
implemen
& throws
the i/p
constru
age and
>= fa

import
class
{ pu
{
}
}

class
{ pu
pu
{
if
}
}

Date __/__/__
Page _____

7) Write a program that demonstrates handling of exceptions in Inheritance Tree. Create a base class called "Father" & derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age & throws the exception WrongAge() when the i/p age<0. In Son class, implement a constructor that uses both Father & Son's age and throws an exception if son's age is >= father's age.

```
import java.util.Scanner;
class WrongAgeException extends Exception
{  public WrongAgeException (String message)
   {  super (message);
   }
}

class Father
{  public int age;
   public Father (int age) throws WrongAgeException
   {
      if (age <0)
         throw new WrongAgeException ("father age
                                       cannot be negative");
      this.age = age;
      System.out.println (" Age set to " + this.age);
   }
}
```

```
class Son extends Father
{   private int Sonage;
    public Son(int FatherAge, int Sonage) throws
    WrongAge Exception
    {   super(FatherAge);
        if (Sonage < 0)
            throw new NrongAge Exception("son age can't
                                        be negative");
        if (Sonage >= FatherAge)
            throw new WrongAge Exception("Son age cannot
            be greater than or equal to Father age");
        this.Sonage = Sonage;
        System.out.println("Son's age is set to: "+this.
                                        Sonage);
    }
}
public class Main
{   psvm(String[] args)
    {   Scanner sc = new Scanner(System.in);
        try
        {   Sop("Enter father age: ");
            int fatherAge = sc.nextInt();
            Sop("Enter Son age: ");
            int Sonage = sc.nextInt()
            Son son = new Son(fatherAge, Sonage);
        }
        catch(WrongAge Exception e)
        {   Sopln("Exception : " + e.getmessage);
        }
        catch(Exception e)
        {   Sopln("Error " + e.getMessage());
    }
}
}
```

Date __/__/__
Page _____

Output:

Enter Father's age :-50
Enter son's age = 25
Exception Father's age cannot be negative!

20/11/24

Date __/__/__
Page

8) Write a program which creates 2 threads one thread displays "BMS College of Engineering" once every ten seconds & another displays "CSE" every 2 sec.

```
class CollegeThread extends Thread
{ public CollegeThread (String message, int interval)
  { new Thread (() ->
    { try
      { while (true)
        { System.out.println(message);
          Thread.sleep (interval);
        }
      } catch (InterruptedException e)
        { System.out.println ("Thread interrupted."
                              +message);
        }
    }).start();
  }
}

public class MultiThreadingExample
{ public static void main (String args[])
  {
    new CollegeThread ("BMS College of Engineering", 10000);
    new CollegeThread ("CSE", 2000);
  }
}
```

Date __/__/__
Page _____

Output:

BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE