1. **Develop a Java program that prints all real solutions to the quadratic equation ax 2 +bx+c = 0. Read in a, b, c and 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.Scanner;
import java.lang.Math;

public class FindRoots{
public static void main(String args[]){
    Scanner s =new Scanner(System.in);
    System.out.println("Enter Coefficients");
    int a=s.nextInt();
    int b=s.nextInt();
    int c=s.nextInt();
    int d=b*b-4*a*c;
if(a==0 && b==0 && c==0){System.out.println("Invalid Input");
}else{
    if(d>0){
            System.out.println("Two Unique Roots Exists");
            System.out.println("Root 1 = "+ ((-b+Math.sqrt(d))/(2*a)));
            System.out.println("Root 2 = "+ ((-b-Math.sqrt(d))/(2*a)));
            }
    else if(d==0){
            System.out.println("Equal Roots Exist");
            System.out.println("Root = "+(-b/(2*a)));
            }
    else{
            System.out.println("No real Solution Exists");
            System.out.println("Imaginary Root1 = "+(-
b/(2*a))+"+"+(Math.sqrt(Math.abs(d))/(2*a))+"i");
            System.out.println("Imaginary Root1 = "+(-b/(2*a))+"-
"+(Math.sqrt(Math.abs(d))/(2*a))+"i");
}
}
}
}
```

```
PS C:\TISSA\OOJ Lab Programs> java FindRoots
Enter Coefficients
2
-5
2
Two Unique Roots Exists
Root 1 = 2.0
Root 2 = 0.5
PS C:\TISSA\OOJ Lab Programs> java FindRoots
Enter Coefficients
1
-2
1
Equal Roots Exist
Root = 1
PS C:\TISSA\OOJ Lab Programs> java FindRoots
Enter Coefficients
1
1
2
No real Solution Exists
Imaginary Root1 = 0+ i1.3228756555322954
Imaginary Root1 = 0- i1.3228756555322954
PS C:\TISSA\OOJ Lab Programs> java FindRoots
Enter Coefficients
0
0
0
Invalid Input
PS C:\TISSA\OOJ Lab Programs>
```

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

```
import java.util.Scanner;
public class ArraySGPA{
        public static void main(String args[]){
        Scanner s=new Scanner(System.in);
        System.out.println("Name");
        String name=s.nextLine();
        System.out.println("USN");
        String usn=s.nextLine();

        System.out.println("Number of Subjects");
        int n=s.nextInt();
        int[] grade=new int[n];
        int[] credits=new int[n];
        for(int i=0;i<n;i++){
        System.out.println("Grade");
        grade[i]=s.nextInt();
        System.out.println("Credits");
        credits[i]=s.nextInt();

}

        SGPA std1=new SGPA(name,usn,grade,credits);
        System.out.println(std1);
        std1.DisplayGrade();
        std1.SGPACalc();



}
}
class SGPA{
```

```
private String name;
private String usn;
private int[] grade;
private int[] credits;
private int sum=0;
private int totalCredits=0;
public SGPA(String name, String usn, int[] grade, int credits[]){
this.name=name;
this.usn=usn;
this.grade=grade;
this.credits=credits;
}
public void SGPACalc(){
for (int i = 0; i < grade.length; i++) {
            sum = sum + (grade[i] * credits[i]);
            totalCredits += credits[i];
        }
        double sgpa = (double) sum / totalCredits;
        System.out.println("SGPA : " + sgpa);
}
public String toString(){
            return "Name : " +name+ "USN : "+ usn;
}
public void DisplayGrade(){
for(int i=0;i<grade.length;i++){
            System.out.println("Subject"+(1+i)+" : "+ grade[i]);
}
}
}
```

```
PS C:\TISSA\OOJ Lab Programs> javac book.java
PS C:\TISSA\OOJ Lab Programs> java book
Enter number of books
3
Enter book name:
abc
Book Name: abc
Enter price
12.5
Enter number of pages
60
Enter author:
def
Enter book name:
ghff

Book Name:
Enter price
23.8
Enter number of pages
354
Enter author:
sddg
Enter book name:
fdghg

Book Name:
Enter price
56.8
Enter number of pages
45
Enter author:
ssdf
Book Name : abc Author : def Price : 12.5 Number of Pages : 60
Book Name :  Author : sddg Price : 23.8 Number of Pages : 354
Book Name :  Author : ssdf Price : 56.8 Number of Pages : 45
```

```
PS C:\TISSA\OOJ Lab Programs> javac ArraySGPA.java
PS C:\TISSA\OOJ Lab Programs> java ArraySGPA
Name
abc
USN
1bm22cs309
Number of Subjects
3
Grade
10
Credits
3
Grade
9
Credits
4
Grade
10
Credits
4
Name : abcUSN : 1bm22cs309
Subject1 : 10
Subject2 : 9
Subject3 : 10
```

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 and 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;
public class book{
public static void main(String args[]){
/*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 and
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.*/
System.out.println("Enter number of books");
Scanner s=new Scanner(System.in);
int n=s.nextInt();

bookDetails[] bookDet=new bookDetails[n];
for(int i=0;i<n;i++){
System.out.println("Enter book name:");
String space1=s.nextLine();
String bookName = s.nextLine();
System.out.println("Book Name: " + bookName);
System.out.println("Enter price");
double price=s.nextDouble();
System.out.println("Enter number of pages");
```

```
int num_pages=s.nextInt();
System.out.println("Enter author:");
String space2=s.nextLine();
String author = s.nextLine();
bookDet[i]=new bookDetails(bookName,author,price,num_pages);
}
for(int i=0;i<n;i++){
System.out.println(bookDet[i]);
}
}
}
class bookDetails{
private String bookName;
private String author;
private double price;
private int num_pages;
public bookDetails(String bookName, String author, double price, int num_pages ){
this.bookName=bookName;
this.author=author;
this.price=price;
this.num_pages=num_pages;
}
void setBookName(String bookName){this.bookName=bookName;}
void setAuthor(String author){this.author=author;}
void setPrice(double price){this.price=price;}
void setNumPages(int num_pages){this.num_pages=num_pages;}
String getBookName(){return bookName;}
String getAuthor(){return author;}
double getPrice(){return price;}
int getNumPages(){return num_pages;}
public String toString(){return "Book Name : "+ bookName +" Author : "+author+" Price :
"+price+" Number of Pages : "+num_pages;}

}
```

```
PS C:\TISSA\OOJ Lab Programs> javac book.java
PS C:\TISSA\OOJ Lab Programs> java book
Enter number of books
3
Enter book name:
abc
Book Name: abc
Enter price
12.5
Enter number of pages
60
Enter author:
def
Enter book name:
ghff

Book Name:
Enter price
23.8
Enter number of pages
354
Enter author:
sddg
Enter book name:
fdghg

Book Name:
Enter price
56.8
Enter number of pages
45
Enter author:
ssdf
Book Name : abc Author : def Price : 12.5 Number of Pages : 60
Book Name :  Author : sddg Price : 23.8 Number of Pages : 354
Book Name :  Author : ssdf Price : 56.8 Number of Pages : 45
```

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 Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape.**

```
public class ShapeMain{
public static void main(String args[]){
/*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 only the method printArea( ) that prints the area of the given shape.*/
Shape s;
s=new Rectangle(4,5);
s.printArea();
s=new Triangle(3,4);
s.printArea();
s=new Circle(4);
s.printArea();
}
}
```

```
abstract class Shape{
protected int a;
protected int b;
public Shape(int a, int b){
this.a=a;
this.b=b;
}
abstract public void printArea();
}
class Rectangle extends Shape{
public Rectangle(int length, int breadth){super(length,breadth);}
public void printArea(){System.out.println("Rectangle Area : "+(a*b));}
}
class Triangle extends Shape{
public Triangle(int height, int base){super(height,base);}
public void printArea(){System.out.println("Triangle Area : "+(0.5*a*b));}
}
class Circle extends Shape{
public Circle(int radius){super(radius,0);}
public void printArea(){System.out.println("Circle Area : "+(3.14*a*a));}
}
```

```
PS C:\TISSA\OOJ Lab Programs> javac ShapeMain.java
PS C:\TISSA\OOJ Lab Programs> java ShapeMain
Rectangle Area : 20
Triangle Area : 6.0
Circle Area : 50.24
```

5. **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 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, 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.

```
public class BankMain{

public static void main(String args[]){

/*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 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, 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.*/

```
    SavingsAccount savingsAccount = new SavingsAccount("John", "SA123", "Savings", 5000);

        savingsAccount.displayAccountDetails();
        savingsAccount.updateBalance(2000);
        savingsAccount.interestCalculator(5, 2);
        savingsAccount.withdrawal(1000);


        CurrentAccount currentAccount = new CurrentAccount("Alice", "CA456", "Current", 2500);


        currentAccount.displayAccountDetails();
        currentAccount.updateBalance(1000);
        currentAccount.withdrawal(500);


        CurrentAccount currentAccountWithPenalty = new CurrentAccount("Bob", "CA789", "Current", 1500);
        currentAccountWithPenalty.displayAccountDetails();
    }
```

```
}
class Account{
double balance;
double deposit;
String name;
String accNo;
String type;
public Account(String name,String accNo, String type,double balance){
this.name=name;
this.accNo=accNo;
this.type=type;
this.balance=balance;
};
public void displayAccountDetails() {
    System.out.println("Account Number: " + accNo);
    System.out.println("Account Type: " + type);
    System.out.println("Customer Name: " + name);
    System.out.println("Balance: " + balance);
}
public void withdrawal(double withdrawalAmount){
if(withdrawalAmount<=balance){
balance=balance-withdrawalAmount;
System.out.println("Balance : "+balance);
}else{
System.out.println("Insufficient Funds");}
}

public void updateBalance(double deposit){
balance+=deposit;
System.out.println("Balance : "+balance);}
}

class SavingsAccount extends Account{
double interest;
```

```
double deposit;
double amount;
 public SavingsAccount(String name, String accNo, String type, double balance) {
        super(name, accNo, type, balance);
    }
public void interestCalculator(double rate, int time){
interest=deposit*rate*time/100;
amount=deposit+interest;
System.out.println("Interest : "+interest);
System.out.println("Amount : "+amount);
}

public boolean chequeFacility(){
return false;}
}

class CurrentAccount extends Account{
public CurrentAccount(String name, String accNo, String type, double balance) {
super(name, accNo, type, balance);
        if (balance <= 2000) {
            System.out.println("Minimum Balance not maintained. Service charge imposed.");
            balance -= 300;
        }
    }
public boolean chequeFacility(){return true;}

}
```

```
PS C:\TISSA\OOJ Lab Programs> javac BankMain.java
PS C:\TISSA\OOJ Lab Programs> java BankMain
Account Number: SA123
Account Type: Savings
Customer Name: John
Balance: 5000.0
Balance : 7000.0
Interest : 0.0
Amount : 0.0
Balance : 6000.0
Account Number: CA456
Account Type: Current
Customer Name: Alice
Balance: 2500.0
Balance : 3500.0
Balance : 3000.0
Minimum Balance not maintained. Service charge imposed.
Account Number: CA789
Account Type: Current
Customer Name: Bob
Balance: 1500.0
```

6. **Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. 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 which has the class External which is a derived class of Student. This class has an array that stores the 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 SEE;
public class External extends CIE.Student{
public int extMarks[];
public External(String name, String usn, String sem, int n){
super(name,usn,sem);
extMarks = new int[n];
}
}
package CIE;
public class Internal{
public int[] marks;
public Internal(int n){marks=new int[n];}
}
package CIE;
public class Student{
protected String name;
protected String usn;
protected String sem;
public Student(String name,String usn,String sem){
this.name=name;
this.usn=usn;
this.sem=sem;}
```

```
}
import java.util.Scanner;
import CIE.Internal;
import SEE.External;
public class packageExample{
public static void main(String args[]){
/*Create a package CIE which has two classes- Student and Internals. The class
Personal
has members like usn, name, sem. 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 which has the class External which is a derived class of Student.
This class has an array that stores the 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.*/
System.out.println("Enter number of students : ");
Scanner s = new Scanner(System.in);
int n=s.nextInt();
int[][] finalMarks=new int[n][5];
for(int i=0;i<n;i++){
System.out.println("Name : ");
String name=s.nextLine();
String space1=s.nextLine();
System.out.println("USN : ");
String usn=s.nextLine();
String space2=s.nextLine();
System.out.println("Sem : ");
String sem=s.nextLine();
String space3=s.nextLine();
Internal internals=new Internal(5);
External externals=new External(name,usn,sem,5);
System.out.println("Enter Internal Marks");
for(int j=0;j<5;j++){System.out.println((j+1));internals.marks[j]=s.nextInt();}
System.out.println("Enter External Marks");
for(int j=0;j<5;j++){System.out.println((j+1));externals.extMarks[j]=s.nextInt();}
for(int j=0;j<5;j++){finalMarks[i][j]=internals.marks[j]+externals.extMarks[j];}
for(int j=0;j<5;j++){System.out.println("Final Marks for Subject "+(j+1)+"
"+finalMarks[i][j]);}
}
}
}
```

```
PS C:\TISSA\OOJ Lab Programs> javac -d . Internal.java
PS C:\TISSA\OOJ Lab Programs> javac -d . External.java
PS C:\TISSA\OOJ Lab Programs> javac packageExample.java
PS C:\TISSA\OOJ Lab Programs> java packageExample
Enter number of students :
2
Name :
std1
USN :
usn1

Sem :
iii

Enter Internal Marks
1
45
2
34
3
50
4
35
5
23
Enter External Marks
1
12
2
25
3
28
4
50
5
23
Final Marks for Subject 1 57
Final Marks for Subject 2 59
Final Marks for Subject 3 78
Final Marks for Subject 4 85
```

7. **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 a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that cases both father and son’s age and throws an exception if son’s age is >=father’s age.**

```
import java.util.Scanner;
public class exceptionExample{
public static void main(String args[]){
```

```
Scanner scanner=new Scanner(System.in);
int fatherAge=scanner.nextInt();
int sonAge=scanner.nextInt();
try{
Father f=new Father(fatherAge);
}catch(WrongAge e){
System.out.println("Invalid Age of Father");}
try {
Son s = new Son(fatherAge, sonAge);
System.out.println("No Errors");
} catch (WrongAge e) {
System.out.println(e.getMessage());
}

}
}
class Father{
private int fAge;
public Father(int fAge)throws WrongAge{
if(fAge<=0) {
throw new WrongAge();
}else{
this.fAge=fAge;
}
}
}

class Son extends Father{
private int sAge;
public Son(int fAge,int sAge) throws WrongAge{
super(fAge);
if(sAge>=fAge){
throw new WrongAge("Son's age should be less than father's age");
}else{
this.sAge=sAge;
}
 }
}

class WrongAge extends Exception {
    public WrongAge() {
        super("Invalid Age");
    }
public WrongAge(String str){
super(str);}

   }
```

```
PS C:\TISSA\OOJ Lab Programs> javac exceptionExample.java
PS C:\TISSA\OOJ Lab Programs> java exceptionExample
12
20
Son's age should be less than father's age
PS C:\TISSA\OOJ Lab Programs> java exceptionExample
0
12
Invalid Age of Father
Invalid Age
PS C:\TISSA\OOJ Lab Programs> java exceptionExample
32
2
No Errors
PS C:\TISSA\OOJ Lab Programs>
```

8. **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.**

```
public class ThreadMain{
public static void main(String args[]){
ThreadOne t1=new ThreadOne();
ThreadTwo t2=new ThreadTwo();
t1.start();
t2.start();
}
}
class ThreadOne extends Thread{
    public void run(){
    try{for(int
i=0;i<5;i++){System.out.println("BMSCE");Thread.sleep(2000);}System.out.println("Threa
d 1 Complete");
}catch(InterruptedException e){
System.out.println("Thread One Interrupted");}
}
}
class ThreadTwo extends Thread{
    public void run(){
    try{for(int
i=0;i<15;i++){System.out.println("CSE");Thread.sleep(500);}System.out.println("Thread 2
Complete");
}catch(InterruptedException e){
System.out.println("Thread Two Interrupted");}
}
}
```

```
PS C:\TISSA\OOJ Lab Programs> java ThreadMain
BMSCE
CSE
CSE
CSE
CSE
BMSCE
CSE
CSE
CSE
CSE
BMSCE
CSE
CSE
CSE
CSE
```

9. **Create a button and add a action listener for Mouse click.**

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

public class EventHandling extends WindowAdapter implements ActionListener {

    Frame f;
    TextField tf;

    EventHandling() {
        f = new Frame("Event Handling Example");

        f.addWindowListener(this);
        tf = new TextField();

        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(tf);

        f.setSize(300, 300);
        f.setLayout(null);
        f.setVisible(true);
    }
```

```
    public void actionPerformed(ActionEvent e) {
      tf.setText("Welcome");
    }

    public void windowClosing(WindowEvent e) {
      f.dispose();
    }

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

10. Programs on IO

```
  import java.io.*;

  public class ByteArrayInput {
    public static void main(String[] args) throws IOException {

      byte[] buf = { 35, 36, 37, 38 }; // byte array

      // Create the new byte array input stream
      ByteArrayInputStream byt = new ByteArrayInputStream(buf);

      int k;
      while ((k = byt.read()) != -1) {
        // Conversion of a byte into character
        char ch = (char) k;
        System.out.println("ASCII value of Character is:" + k + "; Special character is:" + ch);
      }
    }
  }
```

```
  import java.io.FileInputStream;
  import java.io.IOException;

  public class FileEx {
    public static void main(String a[]) throws IOException {
      FileInputStream fin = new FileInputStream("Example.txt");

      System.out.println("Remaining bytes that can be read: " + fin.available());

      int content;
      while ((content = fin.read()) != -1) {
        // Print the character
        System.out.print((char) content);

        // Print the integer value of the character (ASCII value)
        System.out.print(" (" + content + ") ");
      }

      fin.close();
    }
```

```
}


import java.io.FileInputStream;
import java.io.IOException;

public class FileEx2 {
    public static void main(String[] args) throws IOException {

        FileInputStream fin = new FileInputStream("Example.txt");
        byte[] bytes = new byte[20]; // Array to store bytes read from the file

        int numBytesRead = 0;
        int ch;

        while ((ch = fin.read()) != -1) {
            bytes[numBytesRead] = (byte) ch; // Store the byte in the array
            numBytesRead++;
        }

        System.out.println("Number of bytes read: " + numBytesRead);

        System.out.print("Bytes read: ");
        for (int i = 0; i < numBytesRead; i++) {
            System.out.print(bytes[i] + " "); // Print the byte value
        }

        System.out.println(); // Print a newline

        System.out.print("Characters: ");
        for (int i = 0; i < numBytesRead; i++) {
            System.out.print((char) bytes[i] + " "); // Print the character corresponding to the
byte value
        }

        fin.close();
    }
}
```