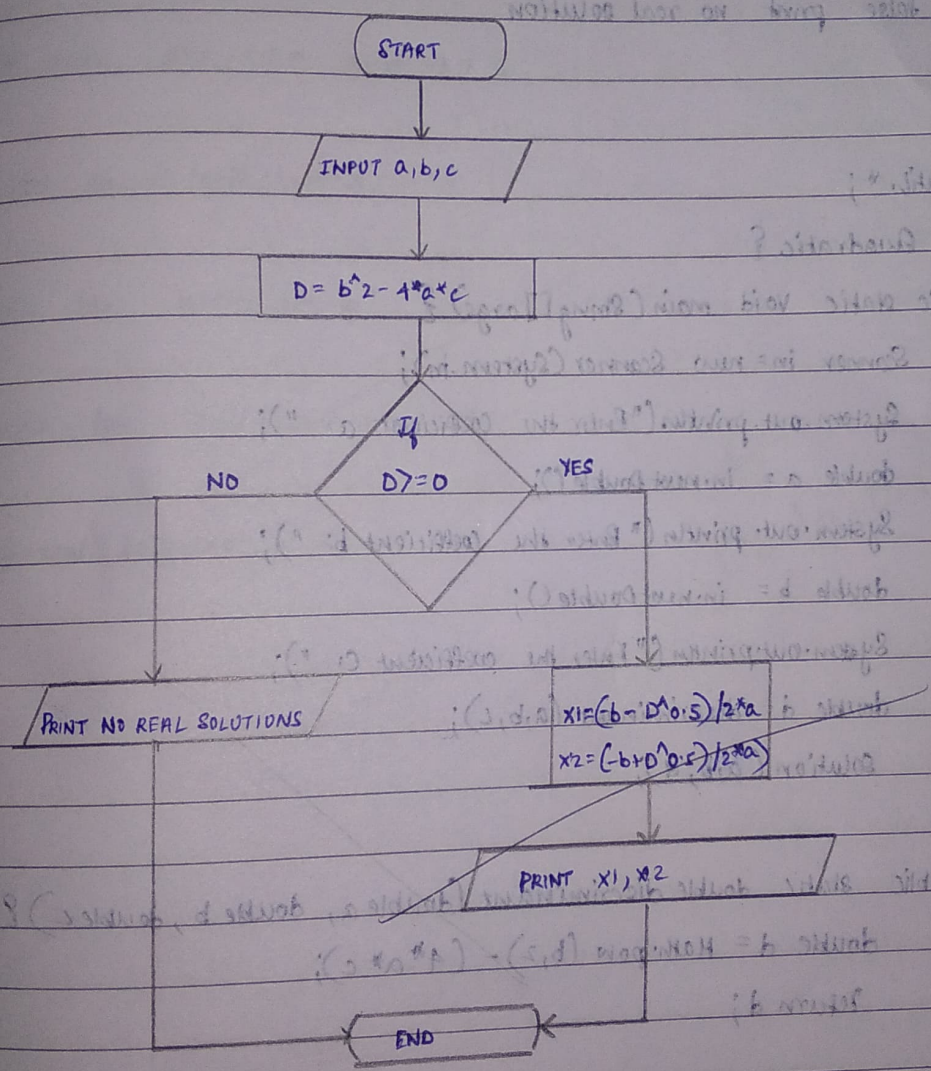
Program1 : Develop a Java program that prints all real solutions to the quadratic equations $ax2+bx+c=0$. Read in a, b, c and use the quadratic formula. If the discriminant $b2-4ac$ is negative, display a message stating that there are no real solutions.

Flowchart



Algorithm

Step 1: Read a, b, c coefficients of quadratic equation

Step 2: Calculate discriminant D= b^2 - 4*a*c

Step 3: Check if discriminant >= 0

Step 4: If true calculate roots x1 and x2 = (-b ± b^0.5)/2*a

Print roots

If false print no real solution

Code:

```
import java.util.*;
public class Quadratic {
    public static void main(String[] args) {
        Scanner in= new Scanner(System.in);
        System.out.println("Enter the coefficient a: ");
        double a = in.nextDouble();
        System.out.println("Enter the coefficient b: ");
        double b = in.nextDouble();
        System.out.println("Enter the coefficient c: ");
        double d = discriminant(a,b,c);
        solution(a,b,d);
}

    public static double discriminant(double a, double b, double c) {
        double d = Math.pow(b,2) - (4*a*c);
        return d;
    }

    public static void solution(double a, double b, double d) {
        if(d>=0) {
            double x1=(-b+Math.sqrt(d))/(2*a);
            double x2 = (-b- Math.sqrt(d))/(2*a);
            System.out.println("The roots are "+x1+"and"+x2);
        }

    else { System.out.println("No real solutions"); }
    }
}
```

Output :

Enter the coefficient a:

1

Enter the coefficient b:

2

Enter the coefficient c:

1

The roots are -1.0 and -1.0

Enter the coefficient a:

1

Enter the coefficient b:

1

Enter the coefficient c:

1

No real solutions

12/01/24

```
winterwizard@pop-os:~/Desktop/Java Practicals/dv_vedith_varma$ javac Prog1.java
winterwizard@pop-os:~/Desktop/Java Practicals/dv_vedith_varma$ java Prog1
Enter the coefficient a:
1
Enter the coefficient b:
2
Enter the coefficient c:
1
The roots are -1.0 and -1.0
Name: D. V. Vedith Varma USN: 1BM22AI040
winterwizard@pop-os:~/Desktop/Java Practicals/dv_vedith_varma$ javac Prog1.java
winterwizard@pop-os:~/Desktop/Java Practicals/dv_vedith_varma$ java Prog1
Enter the coefficient a:
1
Enter the coefficient b:
1
Enter the coefficient c:
1
No real Solutions
Name: D. V. Vedith Varma USN: 1BM22AI040
winterwizard@pop-os:~/Desktop/Java Practicals/dv_vedith_varma$
```

Program 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.

Algorithm:

Step 1: Create a class: Student

Step 2: Declare members usn, name, array to store marks, array to store credits

Step 3: Create a method void accept() to accept student details

Step 4: Read usn, name, credits[], marks[]

Step 5: Create a method void display() to display student details

Step 6: Create a method to calculate sgpa: sgpa()

Step 7: Check marks if marks>=90 assign grade points 10
if marks>=80 and marks<90 assign grade points 9
if marks>=70 and marks<80 assign grade points 8
if marks>=60 and marks<70 assign grade points 7
and so on until marks>=40 and <50 else assign grade points = 4

Step 8: Calculate sum of credits

Step 9: calculate SGPA: $\sum_{i=0}^{n} marks_i * credits_i / \sum_{i=0}^{n} credits_i$

Step 10: Display SGPA

Step 11: Create a public class

Step 12: Create an object of Student class

Step 14: Invoke methods accept(), display(), sgpa() from Student object.

Program:

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

```
public void accept() {
    Scanner in = new Scanner (System.in);
    System.out.println("Enter your name: ");
    name = in.nextLine();
    System.out.println ("Enter your usn: ");
    usn = in.nextLine();
    System.out.println(" Enter the number of subjects:");
    int n;
    n = in.nextInt();
    marks = new Int[n];
    Credits = new int [n];
    for (int i=0; i<n; i++) {
        System.out.println("Enter the marks of subject "+(i+1)+" : ");
        marks[i] = in.nextInt();
        System.out.println ("Enter credits of subject " + (i+1) + ": ");
        Credits[i] = in.nextInt();
    }
}
public void display() {
    System.out.println ("Name: "+ name + "USN: " + usn);
    for (int i=0; i<marks.length; i++) {
        System.out.println ("The marks of a subject "+(i+1)+": "+ marks[i]);
        System.out.println ("The credit of the subject: " + credits[i]);
    }
}
public void sgpa() {
    double score=0; double sum=0;
    for (int i=0; i<marks.length; i++) {
        double grade;
        if (marks[i]>=90) grade=10;
        elseif (marks[i]>=80 && marks[i]<90) grade=9;
        elseif (marks[i]>=70 && marks[i]<80) grade=8;
```

```
                else if (marks[i] >= 60 && marks[i] < 70) grade = 7;
                else if (marks[i] >= 50 && marks[i] < 60) grade = 6;
                else if (marks[i] >= 40 && marks[i] < 50) grade = 5;
                else { System.out.println ("Backlog! Clear subject");
                        return; }
                sum = sum + credits[i];
                score = score + credits[i] * grade;
            }
            score = score / sum;
            System.out.println (" The SGPA of USN: " + usn + "Name: " + name + " is: " + score);
        }
}

public class Program {
    public static void main (String[] args) {
        Student s = new Student();
        s.accept();
        s.display();
        s.sgpa();
    }
}
```

Output:

Enter Your Name:
Vishnu
Enter your USN:
1BM22AI007
Enter the number of subject:
8
Enter the marks of subject 1:
87
Enter the credits of subject 1:
4
Enter the marks of subject 2:
96
Enter the credits of subject 2:
4
Enter the marks of subject 3:
97
Enter the credits of subject 3:
3
Enter the marks of subject 4:
98
Enter the credits of subject 4:

3

Enter the marks of subject 5:

89

Enter the credits of subject 5:

3

Enter the marks of subject 6:

100

Enter the credits of subject 6:

1

Enter the marks of subject 7:

99

Enter the credits of subject 7:

1

Enter the marks of subject 8:

100

Enter the credits of subject 8:

1

Name: Vishnu USN: 1BM22AI007

The marks of a subject 1: 87

The credits of the subject: 4

The marks of a subject 2: 96

The credits of the subject: 4

The marks of subject 3: 97

The credits of subject: 3

The marks of subject 4: 98

The credits of subject: 3

The marks of subject 5: 89

The credits of subject: 3

The marks of subject 6: 100

The credits of subject: 1

The marks of subject 7: 99

The credits of subject: 1

The marks of subject 8: 100

The credits of subject: 1

The SGPA of USN: 1BM22AI007 Name: Vishnu is 9.65

```
winterwizard@pop-os:~/Desktop/Java Practicals$ javac Prog2.java
winterwizard@pop-os:~/Desktop/Java Practicals$ java Prog2
Enter your name:
Vishnu Kumar
Enter your usn:
1RV22AI100
Enter the number of subject:
8
Enter the marks of subject 1 :
87
Enter credits of subject 1 :
4
Enter the marks of subject 2 :
96
Enter credits of subject 2 :
4
Enter the marks of subject 3 :
97
Enter credits of subject 3 :
3
Enter the marks of subject 4 :
98
Enter credits of subject 4 :
3
Enter the marks of subject 5 :
89
Enter credits of subject 5 :
3
Enter the marks of subject 6 :
100
Enter credits of subject 6 :
1
Enter the marks of subject 7 :
99
Enter credits of subject 7 :
1
Enter the marks of subject 8 :
100
Enter credits of subject 8 :
1
Name: Vishnu Kumar USN: 1RV22AI100
The marks of a subject 1 : 87
The credits of the subject : 4
The marks of a subject 2 : 96
The credits of the subject : 4
The marks of a subject 3 : 97
The credits of the subject : 3
The marks of a subject 4 : 98
The credits of the subject : 3
The marks of a subject 5 : 89
```

```
1
Enter the marks of subject 7 :
99
Enter credits of subject 7 :
1
Enter the marks of subject 8 :
100
Enter credits of subject 8 :
1
Name: Vishnu Kumar USN: 1RV22AI100
The marks of a subject 1 : 87
The credits of the subject : 4
The marks of a subject 2 : 96
The credits of the subject : 4
The marks of a subject 3 : 97
The credits of the subject : 3
The marks of a subject 4 : 98
The credits of the subject : 3
The marks of a subject 5 : 89
The credits of the subject : 3
The marks of a subject 6 : 100
The credits of the subject : 1
The marks of a subject 7 : 99
The credits of the subject : 1
The marks of a subject 8 : 100
The credits of the subject : 1
The SGPA of USN: 1RV22AI100 Name: Vishnu Kumar is : 9.65
Name: D. V. Vedith Varma USN: 1BM22AI040
winterwizard@pop-os:~/Desktop/Java Practicals$
```

Program 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

Algorithm:

Step 1: Create class: Book

Step 2: Declare members name, author, price and number of pages as num-pages

Step 3: Create a constructor public Book(String name, String author, double price, int num-of-pages)

Step 4: Set values of members in constructor.

Step 5: Create the following methods to set the name, author, price and number of pages
void setName(String Name), void setAuthor(String Author)
void setPrice(double price), void setPages(int pages)

Step 6: Create the following methods to return, name, author, price and num-pages
String getName(), String getAuthor(), double getPrice(), double getPages()

Step 7: Create method String toString() to return the details in String format

Step 8: return String containing details in toString()

Step 9: Create Class Program declared Public

Step 10: Read the number of books for which detail has to be entered

Step 11: Declare array books of Book type with n objects

Step 12: Initialize each object in array with new Book()

Step 13: Read name, author, price & num-pages

Step 14: Use set methods of class Book to set the values of members

Step 15: Invoke toString() method in class Book to print details of all books

Code:

```
import java.util.*;
Class Book {
    private String name;
    private String author;
    private double price; private int num-pages;
```

```
public Book(){ }
public Book(String name, String author, double price, int num_pages)
{   this.author = author;
    this.name = name;
    this.price = price;
    this.num_pages = num_pages;
}
public void setName(String Name){
        this.name = name;}
public String getName()
  { return name;}
public void setAuthor(String Author){ this.author = author;}
public String getAuthor(){ return Author;}
public void setPrice(double Price){ this.price = price;}
public void getPrice(){ return price;}
public void setPages(int num_pages){ this.num_pages = num_pages;}
public int getPages(){ return num_pages;}
public String toString(){
        return "Book Name: " + name + " Author: " + author + " Price: " + price
            + " Pages: " + num_pages;}
}

public class Program {
    public static void main(String[] args)
  { Scanner in = new Scanner(System.in);
    System.out.println("Enter number of books: ");
    int n = in.nextInt();
    Book[] books = new Book[n];
    for(int i=0; i<n; i++)
    {   books[i] = new Book();
        System.out.println("Enter book name: ");
        String name = in.next(); in.nextLine();
        System.out.println("Enter book author: ");
```

```
String author = in.next(); in.nextLine();
System.out.println("Enter book price:");
double price = in.nextDouble();
System.out.println("Enter number of pages");
int num_pages = in.nextInt();
books[i].setName(name); books[i].setAuthor(author);
books[i].setPrice(price); books[i].setPages(num_pages);
System.out.println("Book: "+ books[i].getName() + "Author: "+
books[i].getAuthor + " details entered.");
}
for (int i=0; i<n; i++)
{ String s = books[i].toString();
  System.out.println(s);
}
}
}
```

Output

Enter number of books

2

Enter book name:

A

Enter book author:

B

Enter the book price:

234

Enter the number of pages:

568

Book: A Author: B details entered

Enter the name of book:

C

Enter book author:

D

Enter the price of book:

456

Enter number of pages:

590

Book: C Author: D details entered.

Book Name: A Author Name: B Price: 234.0 Pages:

Book Name: C Author Name: D Price: 456.0 Pages:

```
winterwizard@pop-os:~/Desktop/Java Practicals$ javac Prog3.java
winterwizard@pop-os:~/Desktop/Java Practicals$ java Prog3
Enter number of books:
2
Enter the name of book:
A
Enter the author of book:
B
Enter the price of the book:
234
Enter the number of pages:
568
Book: A Author: A details entered.
Enter the name of book:
C
Enter the author of book:
D
Enter the price of the book:
456
Enter the number of pages:
590
Book: C Author: C details entered.
Book Name: A Author Name: B Price: 234.0 Pages: 568
Book Name: C Author Name: D Price: 456.0 Pages: 590
Name: D. V. Vedith Varma USN: 1BM22AI040
winterwizard@pop-os:~/Desktop/Java Practicals$
```

Program 4: Develop a Java program to create an abstract class named Shape that contains two integer 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.

Algorithm:

Step 1: Create an abstract class Shape

Step 2: Declare members int a, int b;

Step 3: Create an abstract method printArea() with no body.

Step 4: Create a class: Rectangle that extends class Shape

Step 5: Declare members int a, int b

Step 6: Create method printArea that calculates area using a & b and prints it, where area = a*b

Step 7: Create a class: Triangle that extends Class Shape

Step 8: Declare members int a, int b

Step 9: Create a constructor Triangle that initializes or sets the values of a & b

Step 10: Create method void PrintArea to calculate & print area given by 0.5*a*b

Step 11: Create a class: Circle that extends shape

Step 12: Declare member int a

Step 13: Create a constructor Circle to initialize a

Step 14: Create void PrintArea() to print area given by area = 3.14 * a * a;

Step 15: Creat public class Program

Step 16: Create objects Circle, triangle, rectange of class Shape

Step 17: Invoke printArea method of each object.

Code:

```
abstract class Shape {
    private int a, b;
    Shape () { }
    abstract void printArea ();
}
```

```
class Rectangle extends Shape {
    private int a, b;
        Rectangle(int a, int b)
    {   this.a = a;
        this.b = b;
    }

    void printArea()
    { System.out.println(" Area of rectangle is: "+(a*b)); }
}

class Triangle extends Shape {
    private int a, b;
    Triangle (int a, int b)
    { this.a = a; this.b = b; }
    void printArea()
    { System.out.println(" Area of triangle is: "+(0.5*a*b)); }
}

class Circle extends Shape {
    private int a;
    Circle (int a)
    {   this.a = a; }
    void printArea()
    { System.out.println(" Area of circle: "+(3.14*a*a)); }
}

public class Program {
    public static void main (String[] args)
    {   Shape r = new Rectangle (3,6);
        Shape t = new Triangle (4,6);
        Shape c = new Circle (5);
        r.printArea();
        t.printArea();
        c.printArea();
    }
}
```

DATE: 12/01/24 PAGE:

Output:

Area of rectangle is: 18.

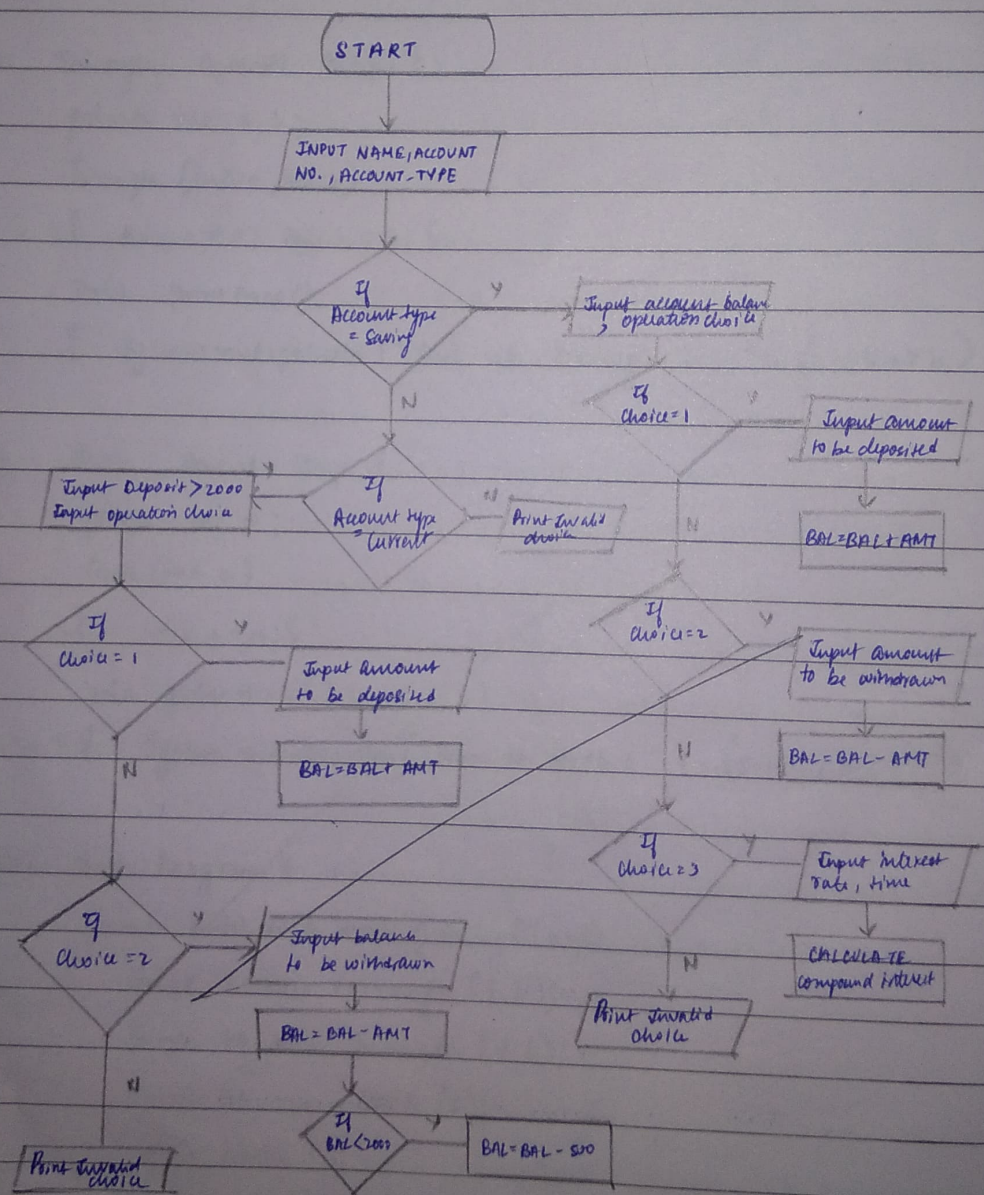
Area of triangle is: 12.0

Area of circle is: 78.5

19/01/23

```
winterwizard@pop-os:~/Desktop/Java Practicals$ javac Prog4.java
winterwizard@pop-os:~/Desktop/Java Practicals$ java Prog4
The area of rectangle is: 18
The area of triangle is: 12.0
The area of circle is: 78.5
Name: D. V. Vedith Varma USN: 1BM22AI040
winterwizard@pop-os:~/Desktop/Java Practicals$
```

Programs: 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 maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores account details and achieve the following: accept deposit & update, display balance, compute & deposit interest, withdraw & update balance.



Algorithm

Step 1: Create a class Account with data members String name, String type, Int account_num, double deposit.

Step 2: Create a method Info to get the above data

Step 3: Create a method details which prints the above data

Step 4: Create a class Savings that extends (inherits) Account

Step 5: Create a method inside Savings deposit(double amount) which deposits amount.

Step 6: Create a methode inside Savings withdraw(double amount) which deducts from balance

Step 7: Create method Interest(int time, int rate)

Step 8: Amount after interest deposit is given by $A = P*(1+r/100)^{*}t$.

Step 9: All methods display balance

Step 10: Create class Current which inherits Account

Step 11: Create method deposit(double amount) which reads balance entered & updates balance

Step 12: Create method withdraw(double amount) which deducts from overall balance and implements check().

Step 13: Create method check() which checks if balance < 5000.

If true balance = balance - 500.

Step 15: Create class Bank

Step 16: Read Name, account type, Account num

Step 17: Input user choice for account type

Step 18: If account type is Savings:

(i) Savings object is created

(ii) details are input through info function inherited by Account

(iii) User choice for methods in Savings class

Step 19: If account type is current:

(i) current object is created

(ii) details input are sent through current object which inherits info.

(iii) user choice for method in current class.

Program:

```
import java.util.Scanner;
class Account {
    String name;
    String type;
    int acc_num;
    double dep;
    public void info (String name, String type, int acc_num, double dep){
        this.name = name;
        this.type = type;
        this.acc_num = acc_num;
        this.dep = dep;
    }
    public details() {
        System.out.println("Name: "+name); System.out.println("Account type: "+ type);
        System.out.println ("Account Number: ") + acc_num);
        System.out.println ("Current Balance:" + dep);
    }
}

class Savings extends Account {
    public void deposit (double amount) {
        dep= dept amount;
        System.out.println("Balance: " + dep);
    }
    public void withdraw (double amount)
    { if(dep< amount)
        { System.out.println("Error Insufficient Funds."); }
        else { dep= dep-amount; }
        System.out.println("Balance: "+dep);
    }
```

```
    public void interest (double t, double r) {
        double dep1 = dep * Math.pow ((1+r/100), t);
        System.out.println ("Interest : " + dep1-dep); dep = dep1;
        System.out.println ("Interest deposited amount : " + dep);
    }
}
class current extends Account {
    public void deposit (double amount) {
        dep = dep + amount;
        System.out.println ("Balance : " + dep);
    }

    public void withdraw (double amount)
    {   if (dep < amount) { System.out.println("Error Insufficient Funds ");
        else { dep = dep - amount; }
        check (dep);
    }

    public void check (double amt)
    {   if (amt < 2000)
        {   if (amt < 500) { dep = 0; }
            else { dep = dep - 500; }
            System.out.println ("Insufficient balance!!! Amount less than Rs 2000. Rs500 deducted");
        } System.out.println ("Balance : " + dep);
    }
}
public class Bank
{   public static void main (String [] args)
    {   Scanner in = new Scanner (System.in); int cl = 1;
        while (cl = 1)
        {   System.out.println ("Enter name : ");
            String name = in.next(); in.nextLine();
            System.out.println ("Enter Account Number : ");
            int acc_no = in.nextInt (); int choice1;
```

```
System.out.println("1. Savings 2. Current");
System.out.println("Enter Account type: ");
choice1 = in.nextInt();
switch (choice1){
    case 1:
        Savings s = new Savings();
        System.out.println("Enter deposit");
        double balance = in.nextDouble();
        s.info(name, type, "Savings", acc_no, balance);
        s.details();
        System.out.println("1. Deposit 2. Withdraw 3. Interest 4. Exit");
        int choice2;
    do{ System.out.println("Enter your choice: ");
        choice2 = in.nextInt();
        switch (choice2){
            case 1:
                System.out.println("Enter amount:");
                double amount1 = in.nextDouble();
                s.deposit(amount1);
                break;
            case 2:
                System.out.println("Enter amount: ");
                double amount2 = in.nextDouble();
                s.withdraw(amount2);
                break;
            case 3:
                System.out.println("Enter time period: ");
                double time = in.nextDouble();
                System.out.println("Enter rate: );
                double rate = in.nextDouble();
                s.interest(time, rate);
                break;
```

```
        case 4:
            break;
        default:
            System.out.println("Invalid choice");
    }while (choice2>=1 && choice2<=3);
    break;
case 2:
    Current c = new current();
    do {
        System.out.println("Enter deposit (>2000)");
        balance = in.nextDouble();
    } while (balance <2000);
    c.info (name, "current", acc_no, balance);
    c.detail();
    System.out.println("1.Deposit 2.withdraw 3.Exit");
    int choice3;   System.out.println("Enter choice"); choice3 = in.nextInt();
                   switch(choice3);
    case 1:
    do { System.out.println("Enter amount: ");
        double amount1 = in.nextDouble();
        c.deposit(amount1); break;
        case 2:
            System.out.println("Enter amount: ");
            double amount2 = in.nextDouble();
            c.withdraw(amount2); break;
        case 3:
            break;
        default:
            System.out.println("Invalid choice.");
    } while (choice3>=1 && choice3 <=2);
    default:
        System.out.println("Invalid choice");
} System.out.println("Enter 1 to continue or 0 to exit");
int c2 = in.nextInt(); } }
```

Output

Enter Name:

Mohan Kumar

Enter Account Number

6746352

Enter Account type:

1

Enter deposit:

7000

Account type: Savings

Account Number: 6746352

Current Balance: 7000.0

1. Deposit 2. Withdraw 3. Interest 4. Exit

Enter your choice:

1

Enter ~~your~~ amount:

1000

Balance: 8000

Enter choice:

3

Enter time period:

3

Enter rate:

7:

Interest: 1800.344

Interest Deposited Amount: 9800.344

Enter choice:

4

Enter 1 to continue or 0 to exit

1

Enter Account type:

2

Enter deposit (>2000)

4500

Name: Rahul

Account type: Current

Account number: 6475786

Current Balance: 4500

1. Deposit 2. Withdraw 3. Exit

Enter choice:

2

Enter amount:

2900

Insufficient Balance !!!

Amount less than 2000

Rs 500 deducted

Balance: 1100.0

Enter choice 3

Enter 1 to continue or 0 to exit

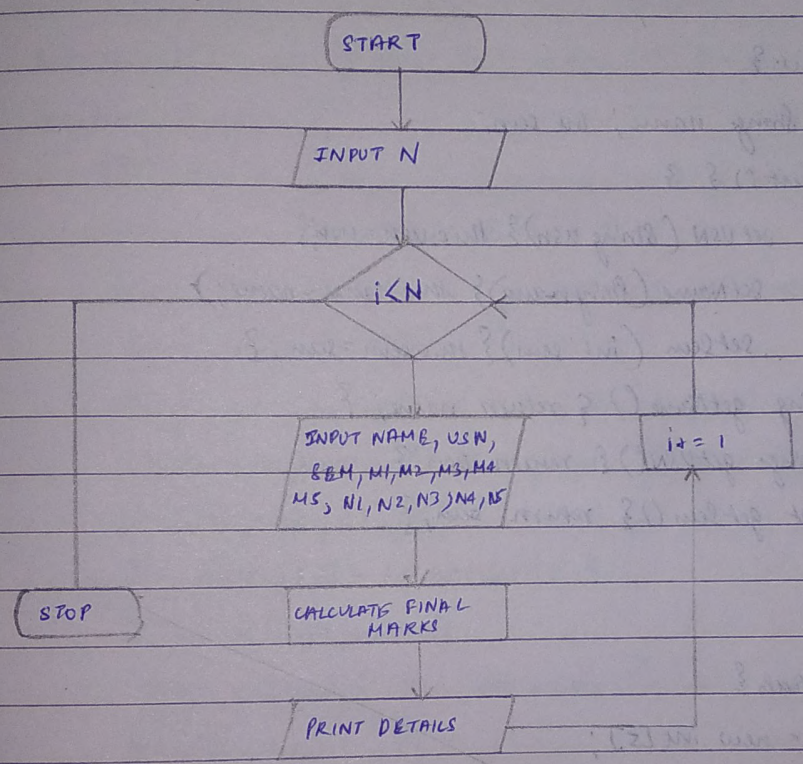
0

02/02/24

```
winterwizard@pop-os:~/Desktop/Java Practicals/dv_vedith_varma$ javac Prog5.java
winterwizard@pop-os:~/Desktop/Java Practicals/dv_vedith_varma$ java Prog5
Enter Name:
Mohan Kumar
Enter Account Number:
1
1.Savings 2.Current
Enter Account Type:
1
Enter deposit
7000
Name: Mohan
Account Type: Savings
Account Number: 1
Current Balance: 7000.0
1.Deposit 2.Withdraw 3.Interest 4.Exit
Enter your choice:
1
Enter amount:
1000
Balance: 8000.0
Enter your choice:
3
Enter time period:
3
Enter rate:
7
Interest: 1800.344000000001
Interest Deposited Amount : 9800.344000000001
Enter your choice:
4
Enter 1 to continue or 0 to exit
1
Enter Name:
Rahul
Enter Account Number:
6475786
1.Savings 2.Current
Enter Account Type:
2
Enter deposit(>2000)
4500
Name: Rahul
Account Type: Current
Account Number: 6475786
Current Balance: 4500.0
1.Deposit 2.Withdraw 3.Exit
Enter your choice:
2
Enter amount:
```

```
Name: Rahul
Account Type: Current
Account Number: 6475786
Current Balance: 4500.0
1.Deposit 2.Withdraw 3.Exit
Enter your choice:
2
Enter amount:
2900
Insufficient Balance!!!! Amount less than Rs. 2000. Rs. 500 deducted. Rs.
Balance: 1100.0
Enter your choice:
3
Invalid Choice
Enter 1 to continue or 0 to exit
0
Name: D. V. Vedith Varma USN: 1BM22AI040
winterwizard@pop-os:~/Desktop/Java Practicals/dv_vedith_varma$
```

Program 6: Create a package CIE which has two classes - Student and Internals. The class Student has members like usn, name, sem. The class internals has an array that stores the internal marks scored in 5 courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of student which has class SEE which stores the marks of 5 courses. Import two packages & calculate final marks of n students.



Algorithm:

→ A package CIE is created for two classes Student and Internal

→ Student class has data members; name, usn and sem and also getter and setter methods for the same.

→ The internal marks has an array to store marks for five courses and getter and setter methods for the same.

→ A package SEE is created which has class External.

→ class external imports internals from CIE and extends it.

→ External has an array to store marks of five students.

→ A class Program is created outside packages to calculate final marks of student

→ The student class imports SEE and CIE packages.

→ No. of Students is need. An array of 'external' class objects is created for n students

→ Name, usn and marks are input for each student. Final marks are calculated and displayed

Program:

```
package CIE;
public class Student {
    String usn; String name; int sem;
    public Student() { }
    public void setUSN (String usn) { this.usn = usn; }
    public void setName (String name) { this.name = name; }
    public void setSem (int sem) { this.sem = sem; }
    public String getName() { return name; }
    public String getUSN() { return usn; }
    public int getSem() { return sem; }
}

package CIE;
public class Internals {
    int[] marks = new int[5];
    public void setMarks (int[] marks) { this.marks = marks; }
    public int[] getMarks() { return marks; }
}

package SEE;
import CIE.Student;
public class Externals extends Student {
    int[] marks = new int[5];
    public Externals() { super(); }
    public void setMarks (int[] marks) { this.marks = marks; }
    public int[] getMarks() { return marks; }
}
```

DATE: 02/02/24 PAGE:

```
import CIE.*;
import SEE.*;
import java.util.*;
public class Program{
    public static void main(String[] args)
  { Scanner in=new Scanner(System.in); System.out.println("Enter number of students:");
    int n= in.nextInt(); int[] marks5= new int[5];
    Externals[] E= new Externals[n]; Internals[] I= new Internals[n];
    for(int i=0; i<n; i++)
    { E[i]= new Externals(); I[i]= new Internals(); int[] marks1=new int[5];
      System.out.println("Enter name: ");
      String name= in.nextLine(); in.nextLine(); E[i].setName(name);
      System.out.println("Enter USN: ");
      String usn= in.nextLine(); E[i].setUSN(usn);
      System.out.println("Enter semester: "); int sem= in.nextInt(); E[i].setSem(sem);
      for(int j=0; j<5; j++)
      { marks1[j]= in.nextInt(); }
      I[i].setMarks(marks1);
      System.out.println("Enter SEE marks: "); int[] marks2= new int[5];
      for(int j=0; j<5; j++)
      { marks2[j]= in.nextInt(); }
      E[i].setMarks(marks2);
      int[] marks3= E[i].getMarks(); int[] marks4= I[i].getMarks();
      System.out.println("Name: "+ E[i].getName() + "USN: "+E[i].getUSN()+ "Semesta: "+
      E[i].getSem());
      System.out.println("The marks of student are: ");
      for(int j=0; j<5; j++)
      { marks5[j]= ((marks3[j]/2)+ marks4[j]); System.out.println(marks5[j]);
    }
  }
}
```

Output :

Enter number of Students :

2

Enter Name :

Ram Kumar

Enter USN :

01

Enter Semester :

2

Enter CIE Marks

50

48

47

44

43

Enter SEE Marks

100

90

99

98

96

Name : Ram Kumar USN : 1 Semester : 2

The marks of student are :

100

93

96

93

91

Enter Name :

Mohan Kumar

Enter USN :

02

Enter Semester :

2

Enter CIE Marks :

50

45

46

44

43

Enter SEE Marks :

88

86

87

90

92

Name : Mohan Kumar USN : 2 Semester : 2

The marks of student are :

94

88

89

89

89

```
winterwizard@pop-os:~/Desktop/Java Practicals/dv_vedith_varma$ javac Prog6.java
winterwizard@pop-os:~/Desktop/Java Practicals/dv_vedith_varma$ java Prog6
Enter number of students:
2
Enter Name:
Ram Kumar
Enter USN:
01
Enter Semester:
2
Enter CIE Marks:
50
48
47
44
43
Enter SEE Marks:
100
90
99
98
96
Name: RamUSN: 01 Semester: 2
The marks of student are:
100
93
96
93
91
Enter Name:
```

```
Enter Name:

Mohan Kumar
Enter USN:
02
Enter Semester:
2
Enter CIE Marks:
50
45
46
44
43
Enter SEE Marks:
88
86
87
90
92
Name: MohanUSN: 02 Semester: 2
The marks of student are:
94
88
89
89
89
Name: D. V. Vedith Varma USN: 1BM22AI040
winterwizard@pop-os:~/Desktop/Java Practicals/dv_vedith_varma$
```

Program 7: Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class "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 inputAge < 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.

Algorithm:

→ A father class has variables father's age and methods checkFage and eprintFage.
→ A constructor to accept father's age is created.
→ The method checkFage throws WrongAge exception
→ It checks if father's age >=0 or age<0. If <0 throws exception
→ A son class is extended from father class with variables age and methods printSage, checkSage which throws exception
→ If in checkSage son's age <= Father's age exception is thrown
→ A custom user defined exception is created
→ In the main function of Prog8 class son's age and father's age is input.

PROGRAM:

```
import java.util.*;
class WrongAge extends Exception {
    public WrongAge(String s) {
        super(s);
    }
}
class Father {
    int f;
    public Father() { ; }
    public Father (int f) { this.f = f; }
    public void checkFage() throws WrongAge {
        try {
            if (f<0) { throw new WrongAge("incorrect father's age"); }
```

```
        catch (WrongAge e) {
            throw new WrongAge("Incorrect father's age");
        }
    }

    public void printFage() {
        System.out.println("Father's age : "+f);
    }
}

class Son extends Father {
    int s, f;
    public Son() { }
    public Son(int f) { super(f); }
    public Son(int s, int f) { super(f); this.s = s; this.f = f;
    }

    public void checkSage() throws WrongAge {
        try { if (s>=f)
                { throw new WrongAge("Incorrect son's age");
                }
            } catch (WrongAge e) {
                throw new WrongAge("Incorrect son's age");
            }

    public void printSage() {
        { System.out.println("Son's age : "+s); }
        }

public class Prog7 {
    public static void main(String [] args) {
        Scanner in = new Scanner(System.in);
        try {
            System.out.println("Enter father's age : ");
            int a1 = in.nextInt();
            Son son1 = new Son(a1);
```

```
            son1.checkFage();
            System.out.println("Enter son's age: ");
            int a2 = in.nextInt();
            Son son2 = new Son(a2, a1);
            son2.checkSage();
            son1.printFage();
            son.printSage();
        } catch (WrongAge e) {
            System.out.println("Caught");
            System.out.println(e.getMessage());
        }
    }
}
```

Output

```
Enter father's age
-1
Caught
Incorrect father's age
Enter father's age
2
Enter son's age
3
Caught
Incorrect son's age
```

```
winterwizard@pop-os:~/Desktop/Java Practicals/dv_vedith_varma$ javac Prog7.java
winterwizard@pop-os:~/Desktop/Java Practicals/dv_vedith_varma$ java Prog7
Enter father's age:
-1
Caught
incorrect father's age
Name: D. V. Vedith Varma USN: 1BM22AI040
winterwizard@pop-os:~/Desktop/Java Practicals/dv_vedith_varma$ java Prog7
Enter father's age:
2
Enter son's age:
3
Caught
incorrect son's age
Name: D. V. Vedith Varma USN: 1BM22AI040
winterwizard@pop-os:~/Desktop/Java Practicals/dv_vedith_varma$
```

PROGRAM 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.

Algorithm

→ A class NewThread is created which implements Runnable.
→ A constructor is created to take in thread name and time for sleep.
→ The value for name of thread is set.
→ Run() function is created
→ It executes a loop for n times and has thread.sleep() to delay the thread execution & prints the name of the thread
→ A class Prog8 is created which has main().
→ Two thread objects with name BMS & CSE are created each with time 10 secs and 2 secs respectively.

PROGRAM:

```
class NewThread implements Runnable {
        String Name;
        Thread t;
        int n;
      NewThread( String threadname, int n){
            Name = threadname;
            this.n = n;
            t = new Thread(this, Name);
            System.out.println("New thread" + t);
            t.start();
        }
      public void run() {
            try { for (int i=0; i<10; i++) {
                  System.out.println(Name + ": " + i);
                  Thread.sleep(n); } }
```

```
        catch (InterruptedException e) {
            System.out.println(Name + "Interrupted");
        }
        System.out.println("Exiting");
    }
}

public class Prog8 {
    public static void main (String[] args) {
        NewThread Ob1 = new NewThread("CSE", 2000);
        NewThread Ob2 = new NewThread("BMS", 10000);
    }
}
```

College of Engineering

Output:

New Thread Thread [#30, CSE, 5, main]

New Thread Thread [#31, BMS, 5, main]

CSE : 0

BMS : 0

CSE : 1

CSE : 2

CSE : 3

CSE : 4

BMS : 1

CSE : 5

CSE : 6

CSE : 7

CSE : 8

CSE : 9

BMS : 2

Exiting

BMS College of Engineering : 3

BMS College of Engineering : 4

BMS College of Engineering : 5

BMS College of Engineering : 6

BMS College of Engineering : 7

BMS College of Engineering : 8

BMS College of Engineering : 9

Exiting

16/2/24

```
winterwizard@pop-os:~/Desktop/Java Practicals/dv_vedith_varma$ javac Prog8.java
winterwizard@pop-os:~/Desktop/Java Practicals/dv_vedith_varma$ java Prog8
New Thread Thread[#20,CSE,5,main]
New Thread Thread[#21,BMS College Of Engineering,5,main]
Name: D. V. Vedith Varma USN: 1BM22AI040
BMS College Of Engineering : 0
CSE : 0
CSE : 1
CSE : 2
CSE : 3
CSE : 4
BMS College Of Engineering : 1
CSE : 5
CSE : 6
CSE : 7
CSE : 8
CSE : 9
BMS College Of Engineering : 2
Exiting
BMS College Of Engineering : 3
BMS College Of Engineering : 4
BMS College Of Engineering : 5
BMS College Of Engineering : 6
BMS College Of Engineering : 7
BMS College Of Engineering : 8
BMS College Of Engineering : 9
Exiting
winterwizard@pop-os:~/Desktop/Java Practicals/dv_vedith_varma$
```

PROGRAM 9: Write a program that creates a user interface to perform integer division. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were zero, the program would throw an ArithmeticException. Display the exception in a message dialog box.

Algorithm:

→ A JFrame is created using java.awt class in SwingDemo() of SwingDemo class
→ Its size and layout is set
→ A JLabel is created to enter divider and divident.
→ A JField is created to enter the input
→ A new button is created.
→ JLabels err, dlab, blab, anslab are created to indicate the errors
→ Buttons, labels and fields created are added to JFrame.
→ An event listener is created
→ The text entered and input in fields is converted to integer.
→ All fields & labels are set to display messages and errors
→ The frame is displayed
→ A main function is created to execute class SwingDemo.

PROGRAM:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo {
    SwingDemo () {
        JFrame jfrm = new JFrame ("Divider App");
        jfrm.setSize (275, 150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JLabel jlab = new JLabel ("Enter the divider and divident");
```

```
JTextField atfj = new JTextField (8);
JTextField btfj = new JTextfield (8);
JButton button= new JButton ("Calculate ");
JLabel err= new JLabel ();
JLabel alab = new JLabel();
JLabel blab = new JLabel();
JLabel anslab = new JLabel();
jfrm.add (jlab);
jfrm.add (atfj);
jfrm.add (btfj);
jfrm.add(button);
jfrm.add (err);
jfrm.add (alab);
jfrm.add (blab);
jfrm.add (anslab);
ActionListener I = new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        System.out.println ("Action event from a text field.");
}    };
atfj.addActionListener (I);
btfj.addActionListener (I);
button.addActionListener (new ActionListener() {
    public void actionPerformed (ActionEvent evt) {
        try {
            int a = Integer.parseInt (atfj.getText());
            int b = Integer.parseInt (btfj.getText());
            int ans= a/b;
            alab.setText ("\nA= "+ a);
            blab.setText ("\nB= "+b);
            anslab.setText ("\nAns= "+ans); err.setText("");
        }
```

```
catch (NumberFormatException e){
        alab.setText("");
        blab.setText("");
        anslab.setText("");
        err.setText("Enter only Integers!");
    }
    catch (ArithmeticException e){
        alab.setText("");
        blab.setText("");
        anslab.setText("");
        err.setText("B should be non zero");
    }
  });
  jfrm.setVisible("true");
}
public static void main(String[]Jargs){
    SwingUtilities.invokeLater(new Runnable(){
        public void run(){
            new SwingDemo();
        }
    });
  }
}
```

Output:

23/2/24

DividerApp	DividerApp
Enter the divider and divident	Enter the divider and divident
8 4	8.4 2
Calculate A=8 B=4 Ans=2	Calculate Enter Only Integers

Divider App Vedith Varma 1BM22AI040

Enter Only Integers! Enter the divider and divident: 8.4

4

Calculate

Divider App Vedith Varma 1BM22AI040

Enter the divider and divident: 8 4

Calculate A = 8 B = 4 Ans = 2

```
winterwizard@pop-os:~/Desktop/Java Practicals/dv_vedith_varma$ javac Prog9.java
winterwizard@pop-os:~/Desktop/Java Practicals/dv_vedith_varma$ java Prog9
Name: D. V. Vedith Varma USN: 1BM22AI040
```

Definition:

1) JFrame class: In Swing functionalities, the container used for an application is called as JFrame. It acts like the main window where components like labels, buttons, text fields are added to create a GUI.
2) JFrm.setSize: Resizes the component so that it has width w and height h.
3) JFrm.setLayout: Used to set layout of the container
4) JFrm.FlowLayout: Used to arrange the components in a line. Default layout of applet or panel.
5) jFrm.setdefaultCloseOperation: Used to specify one of the several option for the close button.
6) JLabel is the class used to create label objects.
7) JTextField is used to create text fields
8) JButton is used to create object of button
9) JFr.add() is used to add objects into JFrame.
10) ActionListener class is used to create ActionListener objects which responds to a specific action.
11) abb.setText(" ") is to set texts into a label.
12) abb.getText(" ") is to obtain texts from a label.
13) jFr.setVisible(true) is to set JFrame object to be visible