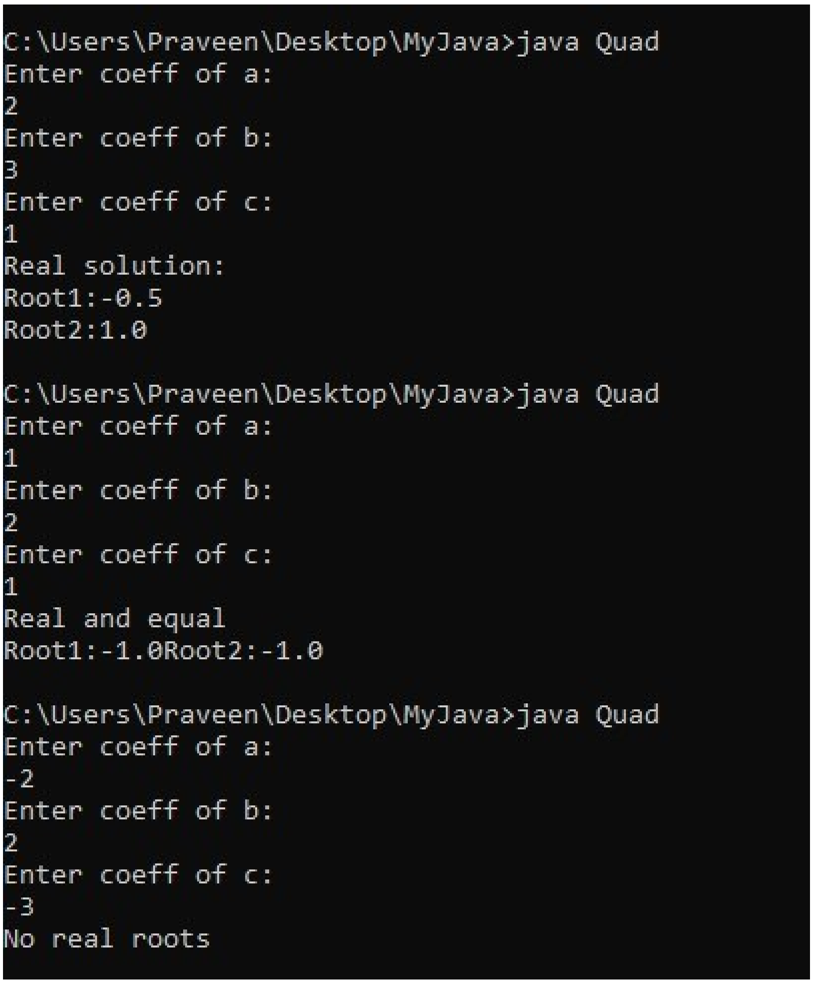
Week 1

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;
public class Quad{
public static void main(String[] args){
double r1,r2;
Scanner sc=new Scanner(System.in);
System.out.println("Enter coeff of a:");
double a=sc.nextDouble();
System.out.println("Enter coeff of b:");
double b=sc.nextDouble();
System.out.println("Enter coeff of c:");
double c=sc.nextDouble();
double disc=(b*b)-(4*a*c);
if(disc<0){
System.out.println("No real roots");
}
else if(disc>0){
r1=(-b+Math.sqrt(disc))/(2*a);
r2=(b+Math.sqrt(disc))/(2*a);
System.out.println("Real solution:");
System.out.println("Root1:"+r1);
System.out.println("Root2:"+r2);
}
else{
r1=r2=(-b/(2*a));
System.out.println("Real and equal");
System.out.println("Root1:"+r1+"Root2:"+r2);
}
}
}
```

Output:



```
C:\Users\Praveen\Desktop\MyJava>java Quad
Enter coeff of a:
2
Enter coeff of b:
3
Enter coeff of c:
1
Real solution:
Root1:-0.5
Root2:1.0

C:\Users\Praveen\Desktop\MyJava>java Quad
Enter coeff of a:
1
Enter coeff of b:
2
Enter coeff of c:
1
Real and equal
Root1:-1.0Root2:-1.0

C:\Users\Praveen\Desktop\MyJava>java Quad
Enter coeff of a:
-2
Enter coeff of b:
2
Enter coeff of c:
-3
No real roots
```

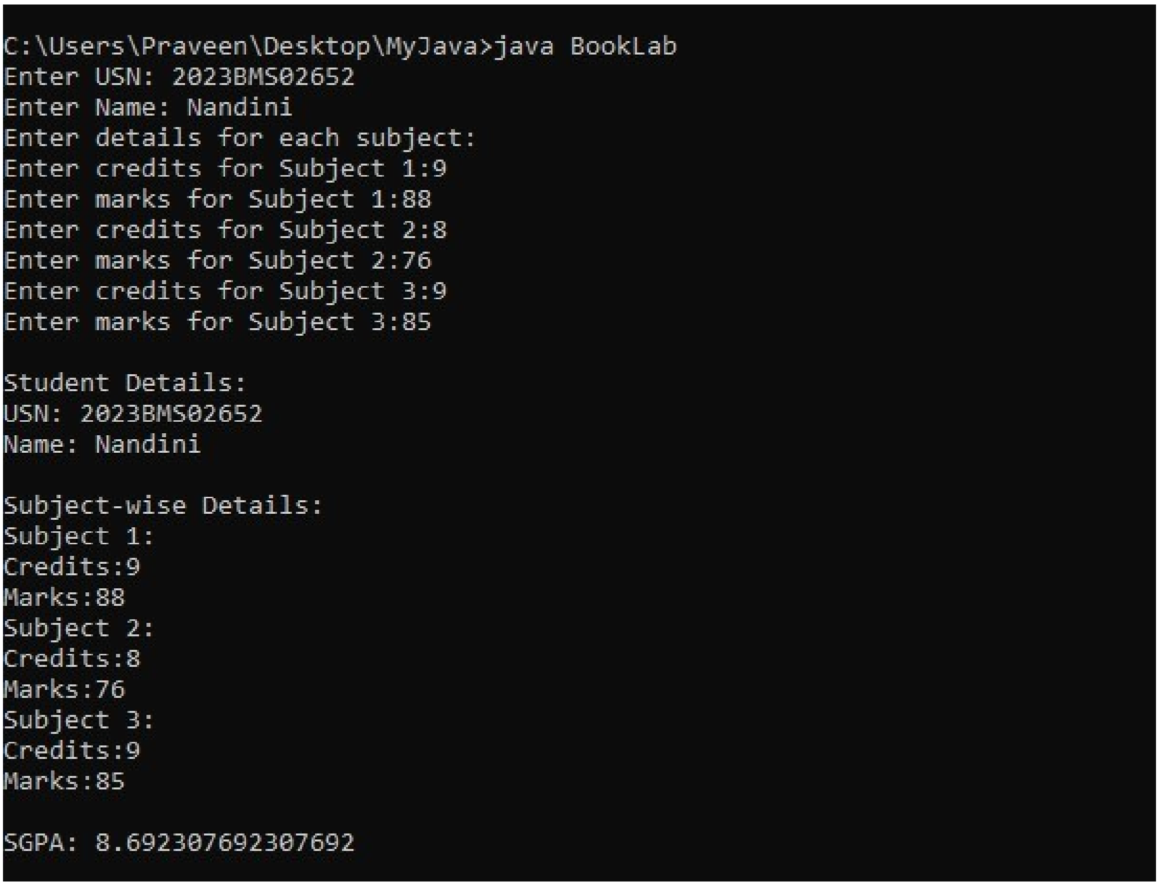
Week 2

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;

 class Student {

 private String usn;

private String name;

 private int[] credits;

private int[] marks;

 public Student(String usn,String name,int numSubjects) {

 this.usn=usn;

this.name=name;

 this.credits=new int[numSubjects];

 this.marks=new int[numSubjects];

}

 public void acceptDetails() {

Scanner sc=new Scanner(System.in);

System.out.print("Enter USN: ");

this.usn=sc.nextLine();

System.out.print("Enter Name: ");

this.name=sc.nextLine();

System.out.println("Enter details for each subject:");

 for(int i=0;i<credits.length;i++) {

System.out.print("Enter credits for Subject "+(i+1)+":");

credits[i]=sc.nextInt();

System.out.print("Enter marks for Subject "+(i+1)+":");

marks[i]=sc.nextInt();

}

}

public void displayDetails(){

System.out.println("\nStudent Details:");
System.out.println("USN: "+usn);
System.out.println("Name: "+name);
System.out.println("\nSubject-wise Details:");
for(int i=0;i<credits.length;i++){
System.out.println("Subject "+(i+1)+":");
System.out.println("Credits:"+credits[i]);
System.out.println("Marks:"+marks[i]);
}
}
public double calculateSGPA() { int totalCredits=0; double
weightedSum=0.0; for(int i=0;i<credits.length;i++) {
totalCredits+=credits[i];
weightedSum+=calculateGradePoints(marks[i])*credits[i];
}
return(weightedSum/totalCredits);
}
private double calculateGradePoints(int marks){
if (marks >= 90) {
return 10.0;
}
 else if (marks >= 80) {
 return 9.0;
}
else if (marks >= 70) {
return 8.0;
}
else if (marks >= 60) {
return 7.0;
}

else if (marks >= 50) {
```

```
return 6.0;
}
else if (marks >= 40) {
return 5.0;
}
 else {return 0.0;}
}
}

public class BookLab {
public static void main(String[] args) {
Student s1 = new Student("123", "John Doe", 3);
s1.acceptDetails();
s1.displayDetails();
double sgpa = s1.calculateSGPA();
System.out.println("\nSGPA: " + sgpa);
}
```

Output:

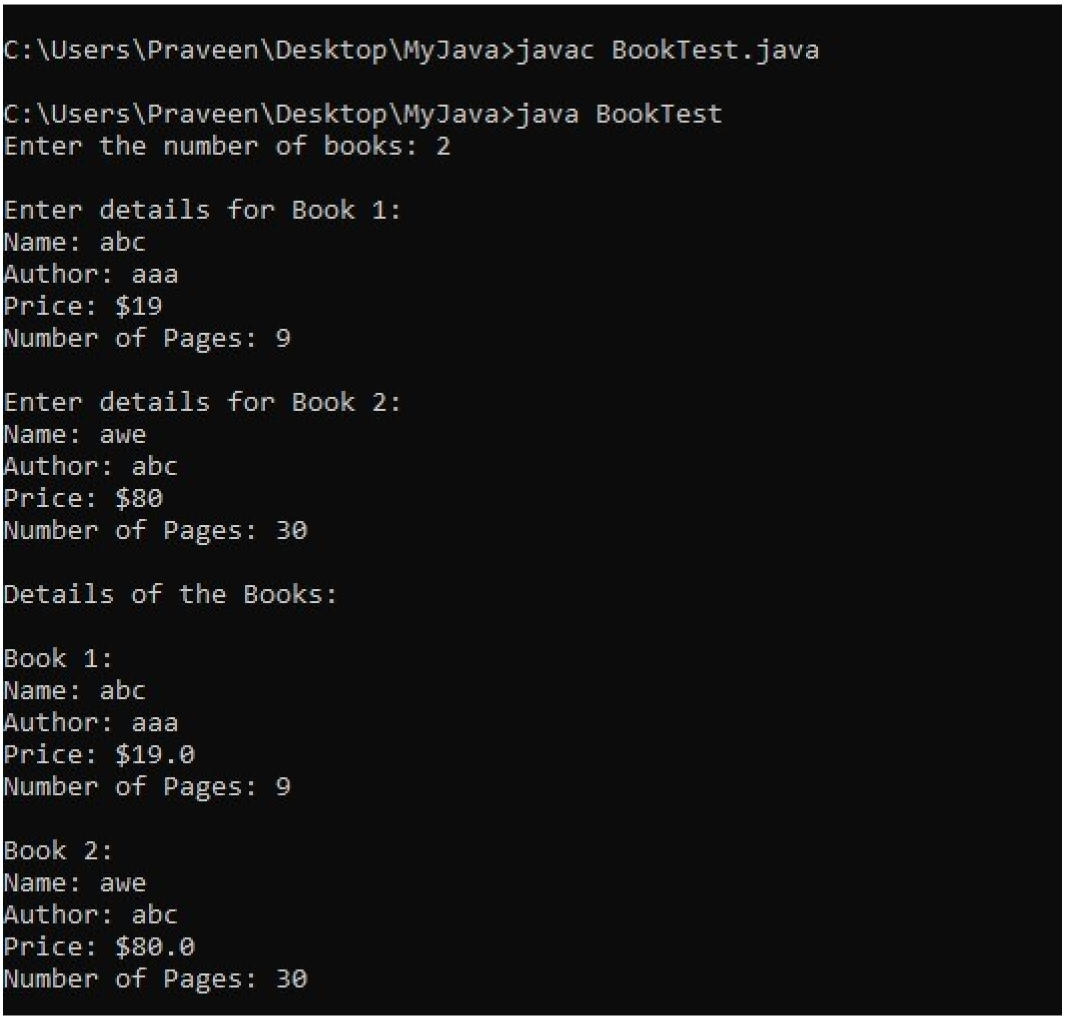


```
C:\Users\Praveen\Desktop\MyJava>java BookLab
Enter USN: 2023BMS02652
Enter Name: Nandini
Enter details for each subject:
Enter credits for Subject 1:9
Enter marks for Subject 1:88
Enter credits for Subject 2:8
Enter marks for Subject 2:76
Enter credits for Subject 3:9
Enter marks for Subject 3:85

Student Details:
USN: 2023BMS02652
Name: Nandini

Subject-wise Details:
Subject 1:
Credits:9
Marks:88
Subject 2:
Credits:8
Marks:76
Subject 3:
Credits:9
Marks:85

SGPA: 8.692307692307692
```

Week 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;
class Book{
private String name;
private String author;
private double price;
private int numPages;
public Book(String name,String author,double price,int
numPages){
this.name = name;
this.author = author;
this.price = price;
this.numPages = numPages;
}
public void setDetails(String name,String author,double price, int
numPages){
this.name = name; this.author = author;
this.price = price;
this.numPages = numPages;
}
public String getDetails(){
return "Name:"+name+"\nAuthor:"+author+"\nPrice:$"+price+"\nNumber of Pages:"+numPages;
}
public String toString() {
return getDetails();
}
}
public class BookTest {
 public static void main(String[]
args) {

Scanner scanner=new Scanner(System.in);

System.out.print("Enter the number of books: ");

int n=scanner.nextInt();

Book1[] books = new

Book1[n];

for(int i=0;i<n;i++){

System.out.println("\nEnter details for Book "+(i+1)+":");

System.out.print("Name: ");

String name=scanner.next();

System.out.print("Author: ");

String author=scanner.next();

System.out.print("Price: $");

Doubleprice=Scanner.nextDouble(
);

System.out.print("Number of

Pages: ");

int numPages=scanner.nextInt();

books[i]=new Book1(name,author,price,numPages);

}

System.out.println("\nDetails of the Books:");

for (int i=0;i<n;i++) {

System.out.println("\nBook "+(i+1)+":\n"+books[i]);

}

}

}
```

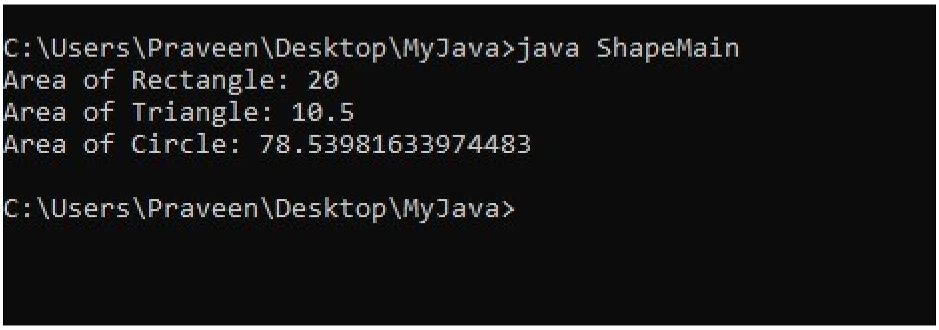


```
C:\Users\Praveen\Desktop\MyJava>javac BookTest.java

C:\Users\Praveen\Desktop\MyJava>java BookTest
Enter the number of books: 2

Enter details for Book 1:
Name: abc
Author: aaa
Price: $19
Number of Pages: 9

Enter details for Book 2:
Name: awe
Author: abc
Price: $80
Number of Pages: 30

Details of the Books:

Book 1:
Name: abc
Author: aaa
Price: $19.0
Number of Pages: 9

Book 2:
Name: awe
Author: abc
Price: $80.0
Number of Pages: 30
```

Week 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.

```
abstract class Shape {
   protected int dimension1;
   protected int dimension2;

   public Shape(int dimension1, int dimension2) {
      this.dimension1 = dimension1;
      this.dimension2 = dimension2;
   }

   public abstract void printArea();
}
class Rectangle extends Shape {
   public Rectangle(int length, int width) {
      super(length, width);
   }

   public void printArea() {
      int area = dimension1 * dimension2;
      System.out.println("Area of Rectangle: " + area);
   }
}
class Triangle extends Shape {
   public Triangle(int base, int height) {
      super(base, height);
   }
public void printArea() {
        double area = 0.5 * dimension1 * dimension2;
        System.out.println("Area of Triangle: " + area);
    }
}
class Circle extends Shape {
    public Circle(int radius) {
        super(radius, 0);}
    public void printArea() {
        double area = Math.PI * dimension1 * dimension1;
        System.out.println("Area of Circle: " + area);
    }
}

public class ShapeMain {
    public static void main(String[] args) {
        Rectangle rectangle = new Rectangle(2,10);
        rectangle.printArea();

        Triangle triangle = new Triangle(3,7);
        triangle.printArea();

        Circle circle = new Circle(5);
        circle.printArea();
    }
}
```



```
C:\Users\Praveen\Desktop\MyJava>java ShapeMain
Area of Rectangle: 20
Area of Triangle: 10.5
Area of Circle: 78.53981633974483

C:\Users\Praveen\Desktop\MyJava>
```

Week 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**

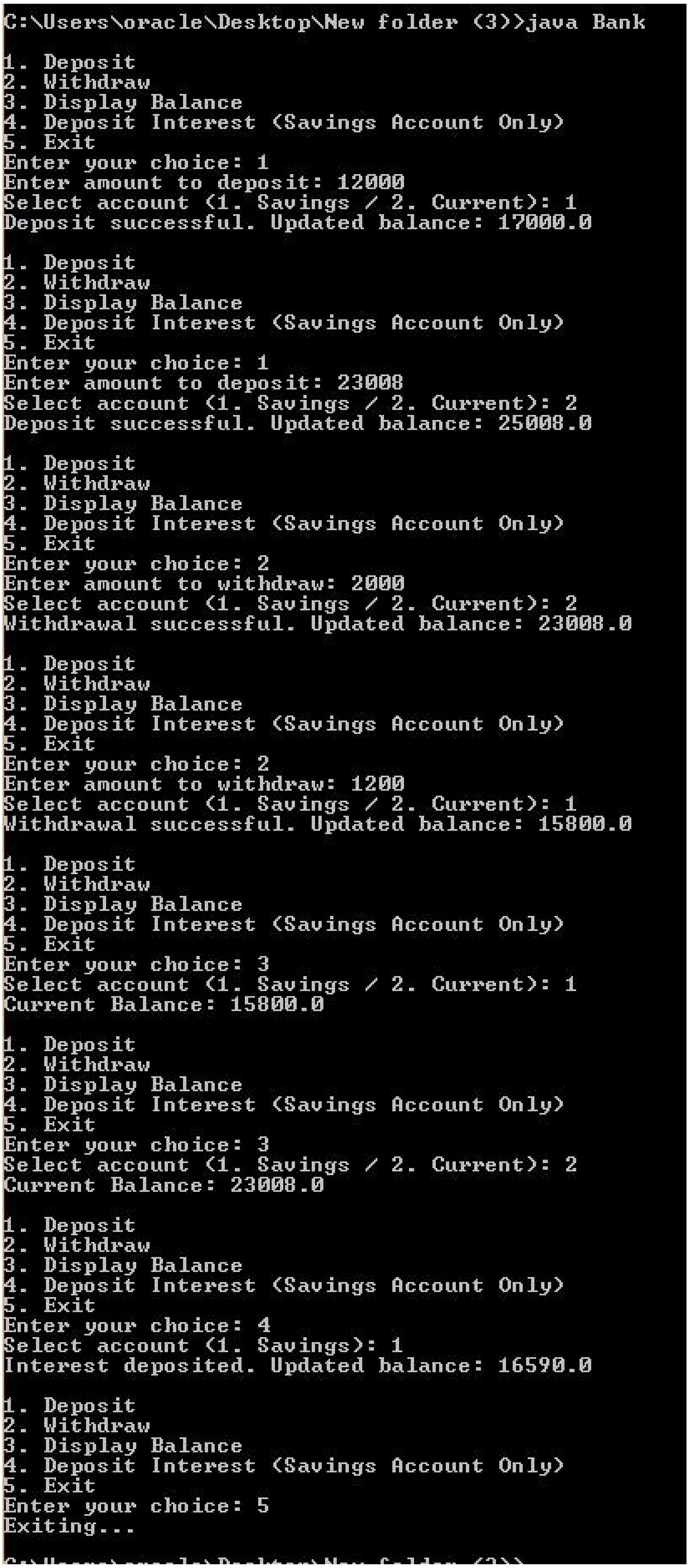
Check for the minimum balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;

class Account {

   String customerName;

   int accountNumber;

   String accountType;

   double balance;

   Account(String name, int accNo, String accType, double bal) {

      customerName = name;

      accountNumber = accNo;

      accountType = accType;

      balance = bal;

   }

   void deposit(double amount) {

      balance += amount;

      System.out.println("Deposit successful. Updated balance: " + balance);

   }

   void displayBalance() {

      System.out.println("Current Balance: " + balance);

   }
```

```
}
class CurAcct extends Account {
    double minBalance = 1000;
    double serviceCharge = 50;
    CurAcct(String name, int accNo, String accType, double bal) {
        super(name, accNo, accType, bal);
    }
    void withdraw(double amount) {
        if (balance - amount >= minBalance) {
            balance -= amount;
            System.out.println("Withdrawal successful. Updated balance: " + balance);
        } else {
            System.out.println("Insufficient balance. Service charge of $" + serviceCharge + " will be applied.");
            balance -= serviceCharge;
            System.out.println("Updated balance after service charge: " + balance);
        }
    }
}
class SavAcct extends Account {
    double interestRate = 0.05;
    SavAcct(String name, int accNo, String accType, double bal) {
        super(name, accNo, accType, bal);
    }
    void depositInterest() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest deposited. Updated balance: " + balance);
    }
    void withdraw(double amount) {
        if (balance - amount >= 0) {
            balance -= amount;
            System.out.println("Withdrawal successful. Updated balance: " + balance);
        } else {
            System.out.println("Insufficient balance.");
        }
    }
}
public class Bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        SavAcct savingsAccount = new SavAcct("John Doe", 1001, "Savings", 5000);
        CurAcct currentAccount = new CurAcct("Jane Smith", 2001, "Current", 2000);
        int choice;
        do {
            System.out.println("\n1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Display Balance");
            System.out.println("4. Deposit Interest (Savings Account Only)");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();
            switch (choice) {
                case 1:
                    System.out.print("Enter amount to deposit: ");
                    double depositAmount = scanner.nextDouble();
                    System.out.print("Select account (1. Savings / 2. Current): ");
                    int accountType = scanner.nextInt();
                    if (accountType == 1)
                        savingsAccount.deposit(depositAmount);
                    else if (accountType == 2)
                        currentAccount.deposit(depositAmount);
                    break;
```

```
case 2:
  System.out.print("Enter amount to withdraw: ");
  double withdrawAmount = scanner.nextDouble();
  System.out.print("Select account (1. Savings / 2. Current): ");
  int accountTypeWithdraw = scanner.nextInt();
  if (accountTypeWithdraw == 1)
    savingsAccount.withdraw(withdrawAmount);
  else if (accountTypeWithdraw == 2)
    currentAccount.withdraw(withdrawAmount);
  break;
case 3:
  System.out.print("Select account (1. Savings / 2. Current): ");
  int accountTypeDisplay = scanner.nextInt();
  if (accountTypeDisplay == 1)
    savingsAccount.displayBalance();
  else if (accountTypeDisplay == 2)
    currentAccount.displayBalance();
  break;
case 4:
  System.out.print("Select account (1. Savings): ");
  int accountTypeInterest = scanner.nextInt();
  if (accountTypeInterest == 1)
    savingsAccount.depositInterest();
  else
    System.out.println("Invalid option.");
  break;
case 5:
  System.out.println("Exiting...");
  break;
default:
  System.out.println("Invalid option. Please try again.");
```

```
            }
        } while (choice != 5);
           scanner.close();
    }
}
```



```
C:\Users\oracle\Desktop\New folder (3)>java Bank

1. Deposit
2. Withdraw
3. Display Balance
4. Deposit Interest (Savings Account Only)
5. Exit
Enter your choice: 1
Enter amount to deposit: 12000
Select account (1. Savings / 2. Current): 1
Deposit successful. Updated balance: 17000.0

1. Deposit
2. Withdraw
3. Display Balance
4. Deposit Interest (Savings Account Only)
5. Exit
Enter your choice: 1
Enter amount to deposit: 23008
Select account (1. Savings / 2. Current): 2
Deposit successful. Updated balance: 25008.0

1. Deposit
2. Withdraw
3. Display Balance
4. Deposit Interest (Savings Account Only)
5. Exit
Enter your choice: 2
Enter amount to withdraw: 2000
Select account (1. Savings / 2. Current): 2
Withdrawal successful. Updated balance: 23008.0

1. Deposit
2. Withdraw
3. Display Balance
4. Deposit Interest (Savings Account Only)
5. Exit
Enter your choice: 2
Enter amount to withdraw: 1200
Select account (1. Savings / 2. Current): 1
Withdrawal successful. Updated balance: 15800.0

1. Deposit
2. Withdraw
3. Display Balance
4. Deposit Interest (Savings Account Only)
5. Exit
Enter your choice: 3
Select account (1. Savings / 2. Current): 1
Current Balance: 15800.0

1. Deposit
2. Withdraw
3. Display Balance
4. Deposit Interest (Savings Account Only)
5. Exit
Enter your choice: 3
Select account (1. Savings / 2. Current): 2
Current Balance: 23008.0

1. Deposit
2. Withdraw
3. Display Balance
4. Deposit Interest (Savings Account Only)
5. Exit
Enter your choice: 4
Select account (1. Savings): 1
Interest deposited. Updated balance: 16590.0

1. Deposit
2. Withdraw
3. Display Balance
4. Deposit Interest (Savings Account Only)
5. Exit
Enter your choice: 5
Exiting...

C:\Users\oracle\Desktop\New folder (3)>
```

Week 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.

//CIE FOLDER

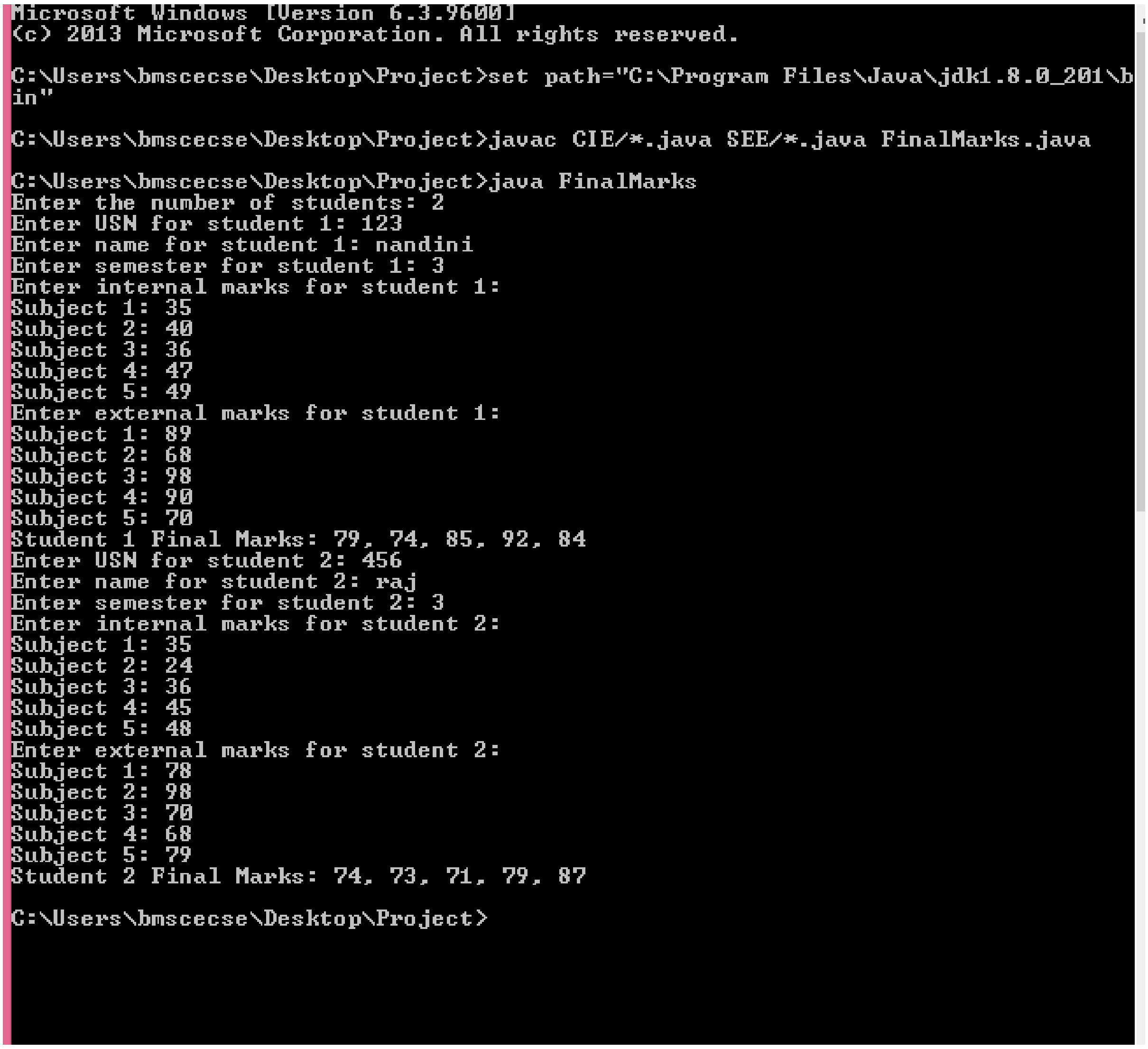
//student class

```
package CIE;
public class Student{
public String usn;
public String name;
public int sem;
public Student(){
this("","",0);
}
public Student(String usn,String name,int sem){
this.usn=usn;
this.name=name;
this.sem=sem;
}
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 getUsn(){
return usn;}
public String getName(){
return name;}
```

```
public String getSem(){
return sem;}
}
}
//internals class
package CIE;
public class Internals{
    private int[] internalMarks = new int[5];
    public Internals() {
    }
public void setInternalMarks(int[] internalMarks) {
        this.internalMarks = internalMarks;
    }
public int[] getInternalMarks() {
        return internalMarks;
    }
}
//SEE FOLDER
//external class
package SEE;
import CIE.Student;
public class External extends Student{
public int[] seeMarks=new int[5];
public External(){
this(" "," "0,new int[5]);
public External(String usn,String name,int sem,int[] seeMarks){
super(usn,name,sem);
this.seeMarks=seeMarks;
}
public void setSeeMarks(int[] seeMarks){
this.seeMarks=seeMarks;
}
public int[] getSeeMarks(){
return seeMarks;
}
}
```

//FINALMARKS CLASS

```
import CIE.Student;
import CIE.Internals;
import SEE.External;
import java.util.Scanner;
public class FinalMarks {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of students: ");
        int n = scanner.nextInt();
        Student[] students = new Student[n];
        Internals[] internals = new Internals[n];
        External[] externals = new External[n];
        for (int i = 0; i < n; i++) {
            students[i] = new Student();
            System.out.print("Enter USN for student " + (i + 1) + ": ");
            students[i].setUsn(scanner.next());
            System.out.print("Enter name for student " + (i + 1) + ": ");
            students[i].setName(scanner.next());
            System.out.print("Enter semester for student " + (i + 1) + ": ");
            students[i].setSem(scanner.nextInt());
            internals[i] = new Internals();
            internals[i].setInternalMarks(inputMarksWithValidation("internal", i, scanner, 0, 50));
            externals[i] = new External(students[i].getUsn(), students[i].getName(), students[i].getSem(), new int[5]);
            externals[i].setSeeMarks(inputMarksWithValidation("external", i, scanner, 0, 100));
            int[] finalMarks = new int[5];
            for (int j = 0; j < 5; j++) {
                finalMarks[j] = internals[i].getInternalMarks()[j] + externals[i].getSeeMarks()[j] / 2;
            }
            System.out.println("Student " + (i + 1) + " Final Marks: " +
                    finalMarks[0] + ", " + finalMarks[1] + ", " + finalMarks[2] + ", " +
                    finalMarks[3] + ", " + finalMarks[4]);
        }scanner.close();
    }
    private static int[] inputMarksWithValidation(String type, int studentIndex, Scanner scanner, int
min, int max) {
        int[] marks = new int[5];
        System.out.println("Enter " + type + " marks for student " + (studentIndex + 1) + ": ");
        for (int i = 0; i < 5; i++) {
            int mark;
            do {
                System.out.print("Subject " + (i + 1) + ": ");
                mark = scanner.nextInt();
                if (mark < 0 || mark > max) {
                    System.out.println("Invalid input. " + type + " marks should be between 0 and " + max + ".
Please try again.");
                }
            } while (mark < 0 || mark > max);
            marks[i] = mark;
        }
        return marks;
    }
}
```

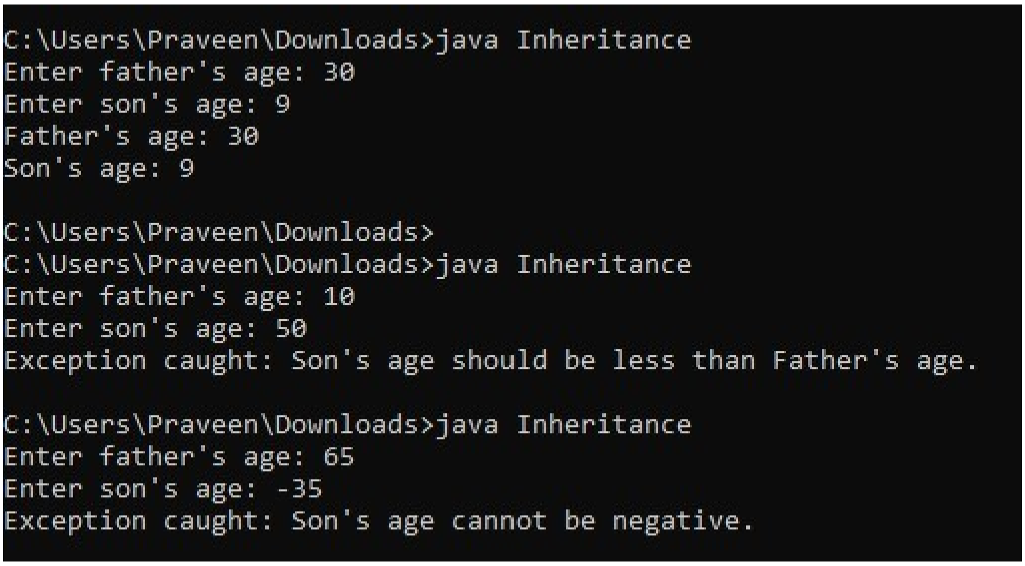


```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\bmscecse\Desktop\Project>set path="C:\Program Files\Java\jdk1.8.0_201\b
in"

C:\Users\bmscecse\Desktop\Project>javac CIE/*.java SEE/*.java FinalMarks.java

C:\Users\bmscecse\Desktop\Project>java FinalMarks
Enter the number of students: 2
Enter USN for student 1: 123
Enter name for student 1: nandini
Enter semester for student 1: 3
Enter internal marks for student 1:
Subject 1: 35
Subject 2: 40
Subject 3: 36
Subject 4: 47
Subject 5: 49
Enter external marks for student 1:
Subject 1: 89
Subject 2: 68
Subject 3: 98
Subject 4: 90
Subject 5: 70
Student 1 Final Marks: 79, 74, 85, 92, 84
Enter USN for student 2: 456
Enter name for student 2: raj
Enter semester for student 2: 3
Enter internal marks for student 2:
Subject 1: 35
Subject 2: 24
Subject 3: 36
Subject 4: 45
Subject 5: 48
Enter external marks for student 2:
Subject 1: 78
Subject 2: 98
Subject 3: 70
Subject 4: 68
Subject 5: 79
Student 2 Final Marks: 74, 73, 71, 79, 87

C:\Users\bmscecse\Desktop\Project>
```

Week 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;

class WrongAge extends Exception {

  public WrongAge(String message) {

    super(message);

  }

}

class Father {

  private int age;

  public Father(int age) throws WrongAge {

    if (age < 0) {

      throw new WrongAge("Age cannot be negative.");

    }

    this.age = age;

  }

  public int getAge() {

    return age;

  }

}

class Son extends Father {

  private int sonAge;

  public Son(int fatherAge, int sonAge) throws WrongAge {

    super(fatherAge);

    if (sonAge < 0) {

      throw new WrongAge("Son's age cannot be negative.");

    }
    if (sonAge >= fatherAge) {
      throw new WrongAge("Son's age should be less than Father's age.");
    }
   this.sonAge = sonAge;
  }
 public int getSonAge() {
    return sonAge;
  }
}
public class Inheritance {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    try {
      System.out.print("Enter father's age: ");
      int fatherAge = scanner.nextInt();
      Father father = new Father(fatherAge);
      System.out.print("Enter son's age: ");
      int sonAge = scanner.nextInt();
      Son son = new Son(fatherAge, sonAge);
     System.out.println("Father's age: " + father.getAge());
      System.out.println("Son's age: " + son.getSonAge());
     } catch (WrongAge e) {
      System.out.println("Exception caught: " + e.getMessage());
    } catch (Exception e) {
      System.out.println("Invalid input. Please enter valid ages.");
    } finally {
      scanner.close();
    }
  }
}
```

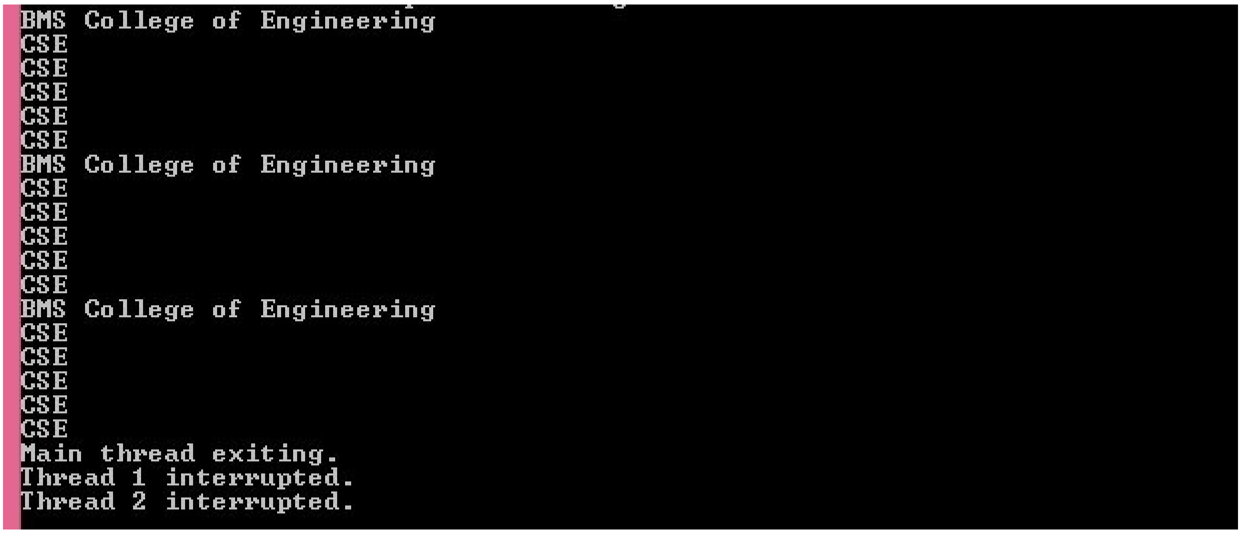


```
C:\Users\Praveen\Downloads>java Inheritance
Enter father's age: 30
Enter son's age: 9
Father's age: 30
Son's age: 9

C:\Users\Praveen\Downloads>
C:\Users\Praveen\Downloads>java Inheritance
Enter father's age: 10
Enter son's age: 50
Exception caught: Son's age should be less than Father's age.

C:\Users\Praveen\Downloads>java Inheritance
Enter father's age: 65
Enter son's age: -35
Exception caught: Son's age cannot be negative.
```

Week 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.

```
class DisplayMessageThread extends Thread {
   private final String message;
   private final long interval;
   DisplayMessageThread(String message, long interval) {
      this.message = message;
      this.interval = interval;
   }
   public void run() {
      try {
         while (true) {
            System.out.println(message);
            Thread.sleep(interval);
         }
      } catch (InterruptedException e) {
         System.out.println(Thread.currentThread().getName() + " interrupted.");
      }
   }
}

public class ThreadMain {
   public static void main(String[] args) {
      DisplayMessageThread thread1 = new DisplayMessageThread("BMS College of Engineering", 10000);
      DisplayMessageThread thread2 = new DisplayMessageThread("CSE", 2000);
      thread1.setName("Thread 1");
      thread2.setName("Thread 2");
      thread1.start();
      thread2.start();
      try {
         Thread.sleep(30000);
      } catch (InterruptedException e) {
         System.out.println("Main thread interrupted.");
      }
      thread1.interrupt();
      thread2.interrupt();
      System.out.println("Main thread exiting.");
   }
}
```



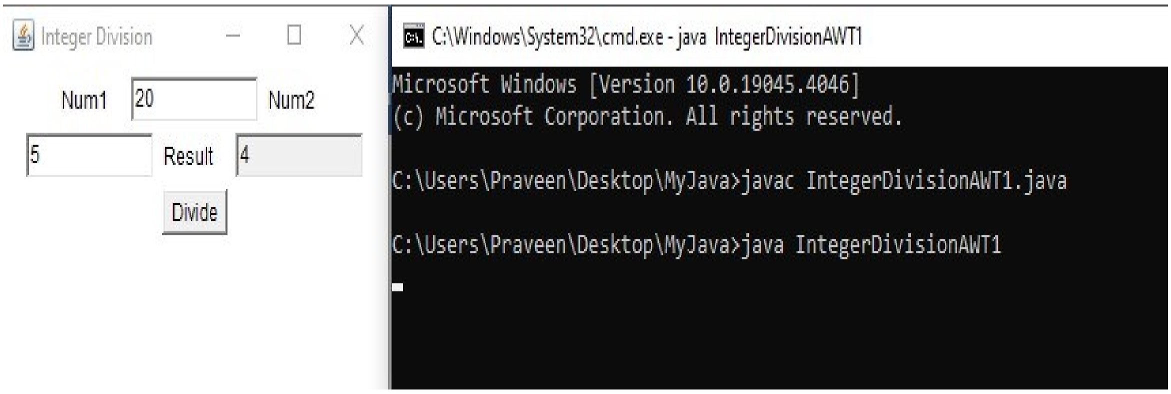
```
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
Main thread exiting.
Thread 1 interrupted.
Thread 2 interrupted.
```

Week 9

Write a program that creates a user interface to perform integer divisions. 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 Arithmetic Exception Display the exception in a message dialog box.

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

public class IntegerDivisionAWT1 extends Frame implements ActionListener {
   private TextField num1Field, num2Field, resultField;
   private Button divideButton;

   public IntegerDivisionAWT1() {
      setTitle("Integer Division");
      setSize(300, 200);
      setLayout(new FlowLayout());
      addWindowListener(new WindowAdapter() {
         public void windowClosing(WindowEvent e) {
            dispose();
         }
      });

      Label num1Label = new Label("Num1");
      num1Field = new TextField(10);
      Label num2Label = new Label("Num2");
      num2Field = new TextField(10);
      Label resultLabel = new Label("Result");
      resultField = new TextField(10);
      resultField.setEditable(false);

      divideButton = new Button("Divide");
      divideButton.addActionListener(this);

      add(num1Label);
      add(num1Field);
      add(num2Label);
      add(num2Field);
      add(resultLabel);
      add(resultField);
      add(divideButton);

      setVisible(true);
   }

   public void actionPerformed(ActionEvent e) {
      if (e.getSource() == divideButton) {
```

```
      try {
        int num1 = Integer.parseInt(num1Field.getText());
        int num2 = Integer.parseInt(num2Field.getText());
        if (num2 == 0) {
          showMessageDialog("Cannot divide by zero");
          return;
        }
        int result = num1 / num2;
        resultField.setText(String.valueOf(result));
      } catch (NumberFormatException ex) {
        showMessageDialog("Please enter valid integers for Num1 and Num2");
      }
    }
  }

  private void showMessageDialog(String message) {
    Dialog d = new Dialog(this, "Error", true);
    d.setSize(300, 100);
    d.setLayout(new FlowLayout());
    Label l = new Label(message);
    d.add(l);
    Button okButton = new Button("OK");
    okButton.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        d.dispose();
      }
    });
    d.add(okButton);
    d.setVisible(true);
  }

  public static void main(String[] args) {
    new IntegerDivisionAWT1();
  }
}
```



Num1 20 Num2 5 Result 4

```
Microsoft Windows [Version 10.0.19045.4046]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Praveen\Desktop\MyJava>javac IntegerDivisionAWT1.java

C:\Users\Praveen\Desktop\MyJava>java IntegerDivisionAWT1
```