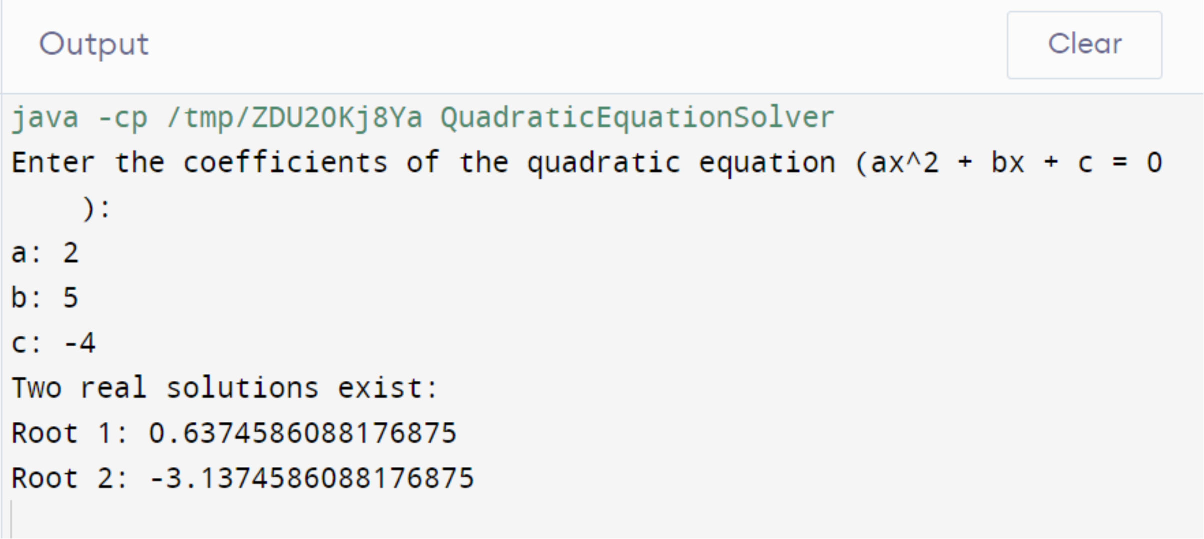
1) Quadratic Equations:

```
import java.util.Scanner;


public class QuadraticEquationSolver {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);


        // Input coefficients a, b, and c
        System.out.println("Enter the coefficients of the quadratic equation (ax^2 + bx + c = 0):");
        System.out.print("a: ");
        double a = scanner.nextDouble();
        System.out.print("b: ");
        double b = scanner.nextDouble();
        System.out.print("c: ");
        double c = scanner.nextDouble();


        // Calculate the discriminant
        double discriminant = b * b - 4 * a * c;


        // Check if there are real solutions
        if (discriminant > 0) {
            // Two real solutions
            double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
            double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
            System.out.println("Two real solutions exist:");
            System.out.println("Root 1: " + root1);
            System.out.println("Root 2: " + root2);
        } else if (discriminant == 0) {
            // One real solution
            double root = -b / (2 * a);
            System.out.println("One real solution exists:");
            System.out.println("Root: " + root);
        } else {
            // No real solutions
            System.out.println("No real solutions exist.");
        }

        scanner.close();
    }
}
```

KEERTHI REDDY

USN: 1BM22CS094

Output:



```
java -cp /tmp/ZDU2OKj8Ya QuadraticEquationSolver
Enter the coefficients of the quadratic equation (ax^2 + bx + c = 0
    ):
a: 2
b: 5
c: -4
Two real solutions exist:
Root 1: 0.6374586088176875
Root 2: -3.1374586088176875
```

2) STUDENT CLASS

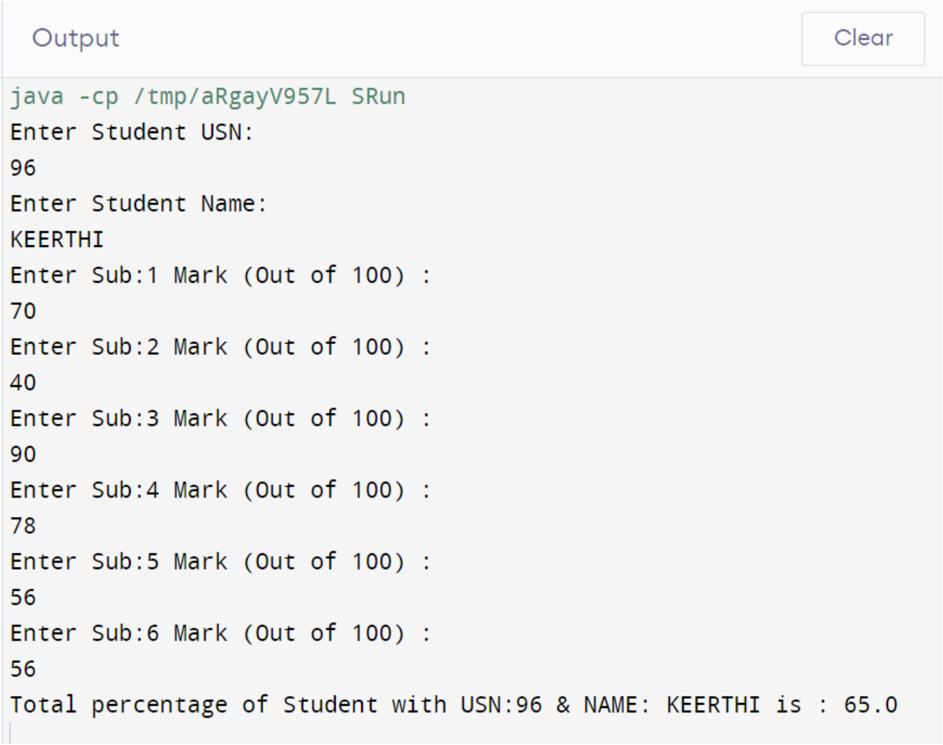
```
import java.util.Scanner;

class Student {
    private String usn;
    private String name;
    private int[] credits;
```

```java
private int[] marks;

// Constructor
public Student(String usn, String name, int numSubjects) {
    this.usn = usn;
    this.name = name;
    this.credits = new int[numSubjects];
    this.marks = new int[numSubjects];
}

// Method to accept details of the student
public void acceptDetails(Scanner scanner) {
    System.out.println("Enter details for student " + name + ":");
    System.out.print("Enter USN: ");
    this.usn = scanner.next();
    System.out.print("Enter Name: ");
    this.name = scanner.next();
    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) + ": ");
        this.credits[i] = scanner.nextInt();
        System.out.print("Enter marks for subject " + (i + 1) + ": ");
        this.marks[i] = scanner.nextInt();
    }
}

// Method to display details of the student
public void displayDetails() {
    System.out.println("USN: " + this.usn);
    System.out.println("Name: " + this.name);
    System.out.println("Subject-wise details:");
    for (int i = 0; i < credits.length; i++) {
        System.out.println("Subject " + (i + 1) + ": Credits - " + credits[i] + ", Marks - " + marks[i]);
        }
    }

    // Method to calculate SGPA of the student
    public double calculateSGPA() {
        int totalCredits = 0;
        double totalGradePoints = 0.0;
        for (int i = 0; i < credits.length; i++) {
            totalCredits += credits[i];
            totalGradePoints += calculateGradePoints(marks[i]) * credits[i];
        }
        return totalGradePoints / totalCredits;
    }

    // Helper method to calculate grade points based on marks
    private double calculateGradePoints(int marks) {
        if (marks >= 90) return 10;
        else if (marks >= 80) return 9;
        else if (marks >= 70) return 8;
        else if (marks >= 60) return 7;
        else if (marks >= 50) return 6;
        else if (marks >= 40) return 5;
        else return 0;
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Accept details of the student
        System.out.print("Enter the number of subjects: ");
        int numSubjects = scanner.nextInt();
        Student student = new Student("", "", numSubjects);
        student.acceptDetails(scanner);

        // Display details of the student
        System.out.println("\nDetails of the student:");
        student.displayDetails();

        // Calculate and display SGPA
        double sgpa = student.calculateSGPA();
        System.out.println("\nSGPA: " + sgpa);

        scanner.close();
    }
}
```

KEERTHI REDDY

USN: 1BM22CS094

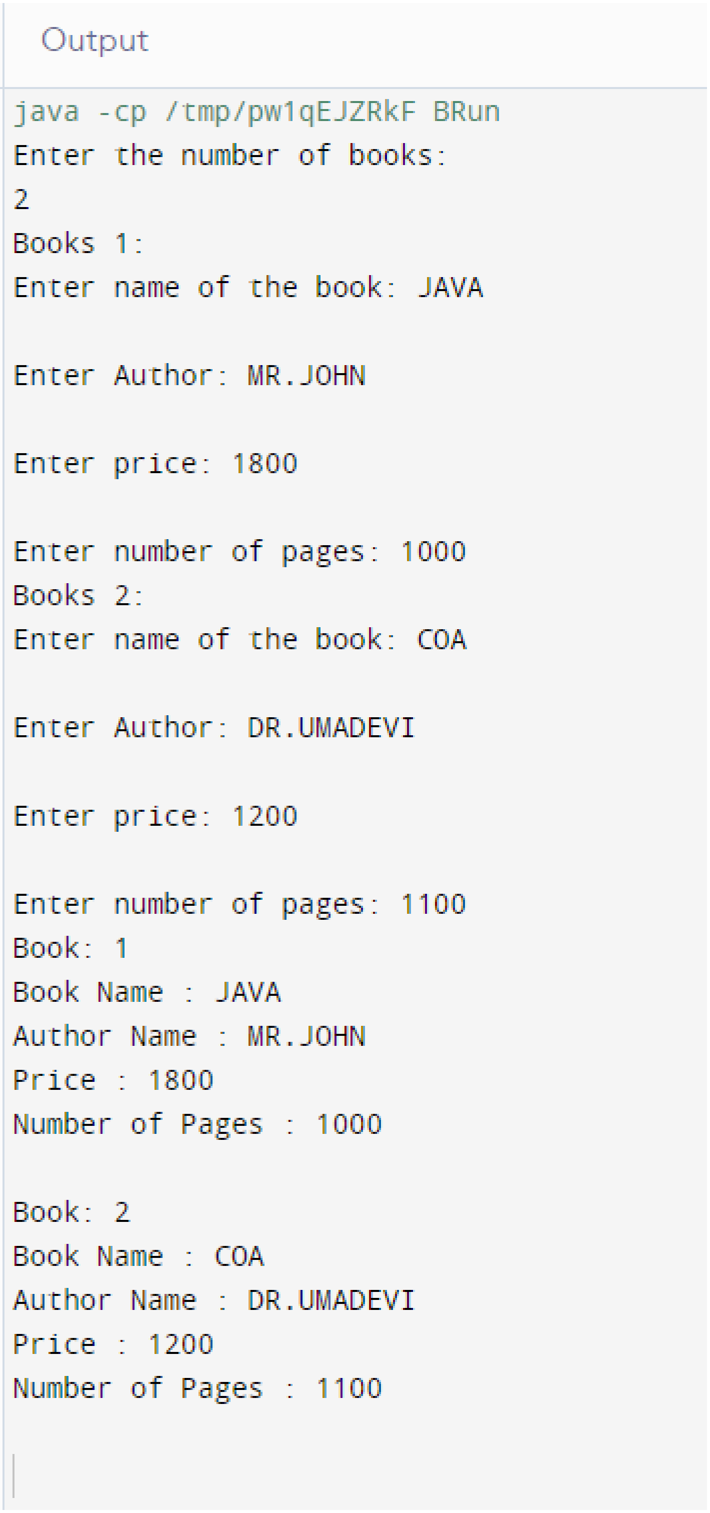
OUTPUT:



```
java -cp /tmp/aRgayV957L SRun
Enter Student USN:
96
Enter Student Name:
KEERTHI
Enter Sub:1 Mark (Out of 100) :
70
Enter Sub:2 Mark (Out of 100) :
40
Enter Sub:3 Mark (Out of 100) :
90
Enter Sub:4 Mark (Out of 100) :
78
Enter Sub:5 Mark (Out of 100) :
56
Enter Sub:6 Mark (Out of 100) :
56
Total percentage of Student with USN:96 & NAME: KEERTHI is : 65.0
```

3) BOOK CLASS

```
import java.util.Scanner;


class Book {
    private String name;
    private String author;
    private double price;
    private int numPages;


    // Constructor to set the values for the members
    public Book(String name, String author, double price, int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }


    // Setter methods
    public void setName(String name) {
        this.name = name;
    }


    public void setAuthor(String author) {
        this.author = author;
    }


    public void setPrice(double price) {
        this.price = price;
    }


    public void setNumPages(int numPages) {
        this.numPages = numPages;
    }
    // Getter methods
    public String getName() {
        return name;
    }

    public String getAuthor() {
        return author;
    }

    public double getPrice() {
        return price;
    }

    public int getNumPages() {
        return numPages;
    }

    // toString method to display complete details of the book
    public String toString() {
        return "Book Details:\nName: " + name + "\nAuthor: " + author + "\nPrice: $" + price + "\nNumber of
Pages: " + numPages;
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of books: ");
        int n = scanner.nextInt();

        // Create an array to hold n book objects
        Book[] books = new Book[n];

        // Input details for each book
        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: $");
            double price = scanner.nextDouble();
            System.out.print("Number of Pages: ");
            int numPages = scanner.nextInt();

            // Create a new Book object with the input details
            books[i] = new Book(name, author, price, numPages);
        }

        // Display details of all the books
        System.out.println("\nDetails of all books:");
        for (int i = 0; i < n; i++) {
            System.out.println("\nBook " + (i + 1) + ":\n" + books[i].toString());
        }

        scanner.close();
    }
}
```



```
java -cp /tmp/pw1qEJZRkF BRun
Enter the number of books:
2
Books 1:
Enter name of the book: JAVA

Enter Author: MR.JOHN

Enter price: 1800

Enter number of pages: 1000
Books 2:
Enter name of the book: COA

Enter Author: DR.UMADEVI

Enter price: 1200

Enter number of pages: 1100
Book: 1
Book Name : JAVA
Author Name : MR.JOHN
Price : 1800
Number of Pages : 1000

Book: 2
Book Name : COA
Author Name : DR.UMADEVI
Price : 1200
Number of Pages : 1100
```

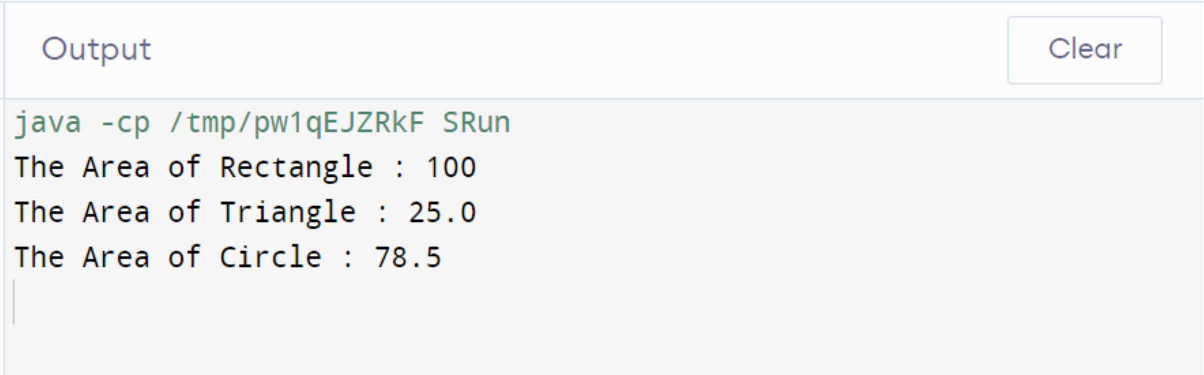
4) SHAPE CLASS

```
abstract class Shape {

    public int side1, side2;
    abstract void printArea();
}


class Rectangle extends Shape {
    Rectangle(int length, int breadth) {
        this.side1 = length;
        this.side2 = breadth;
    }
    void printArea() {
        System.out.println("The Area of Rectangle : " + (side1 * side2));
    }
}

class Triangle extends Shape {
    Triangle(int base, int height) {
        this.side1 = base;
        this.side2 = height;
    }
    void printArea() {
        System.out.println("The Area of Triangle : " + (0.5 * side1 * side2));
    }
}

class Circle extends Shape {
    Circle(int rad) {
        this.side1 = this.side2 = rad;
    }
    void printArea() {
        System.out.println("The Area of Circle : " + (3.14 * side1 * side2));
    }
}
```

```
class SRun{
    public static void main(String[] args) {
        Rectangle r = new Rectangle(10, 10);
        Triangle t = new Triangle(5, 10);
        Circle c = new Circle(5);

        r.printArea();
        t.printArea();
        c.printArea();
    }
}
```

KEERTHI REDDY

USN: 1BM22CS094



```
java -cp /tmp/pw1qEJZRkF SRun
The Area of Rectangle : 100
The Area of Triangle : 25.0
The Area of Circle : 78.5
```

5) BANK CLASS

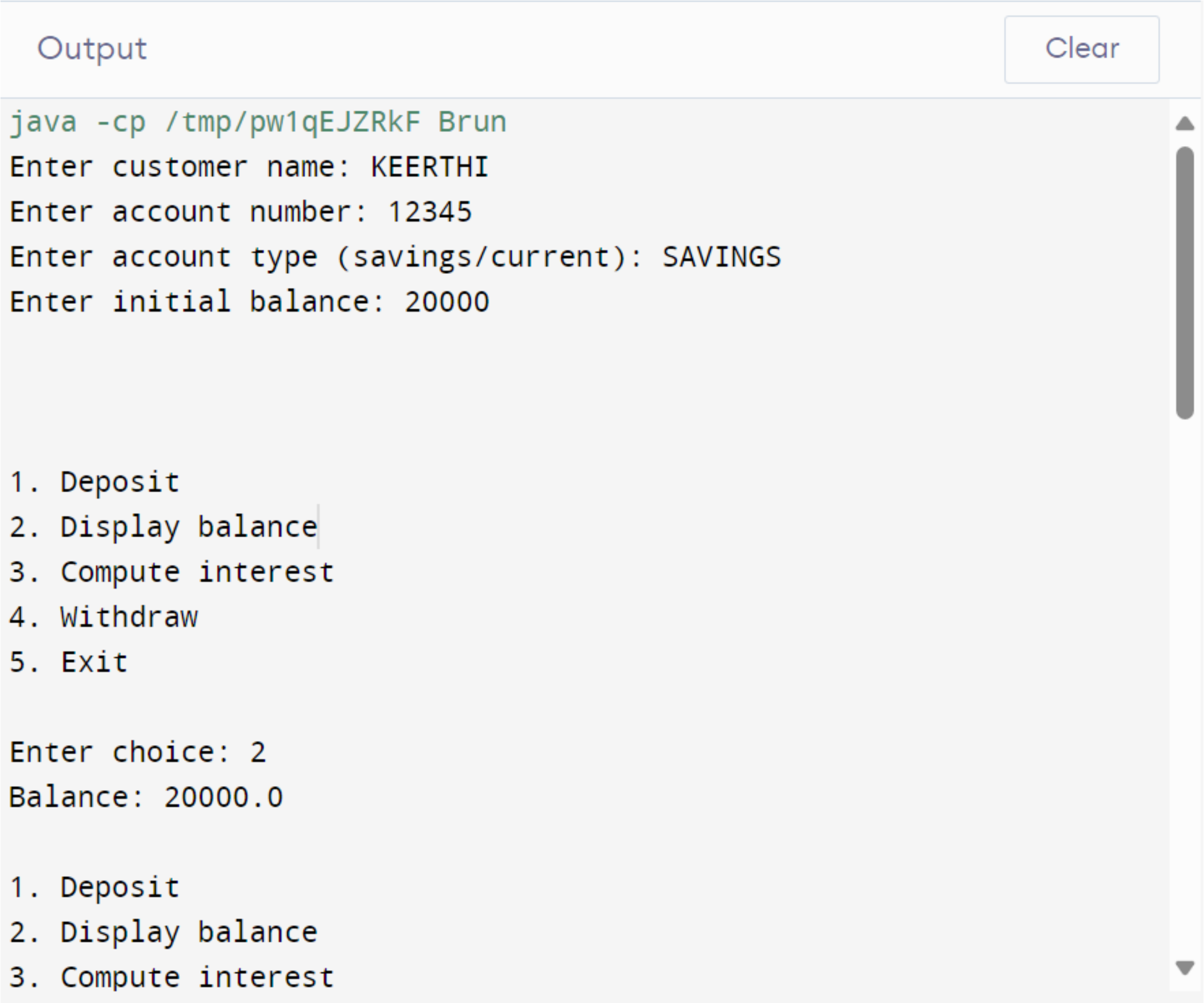
```
import java.util.Scanner;

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

    Account(String customerName, int accountNumber, String accountType, double balance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = balance;
    }

    abstract void deposit(double amount);

    abstract void displayBalance();

    abstract void computeInterest();

    abstract void withdraw(double amount);
}

class SavingsAccount extends Account {
    SavingsAccount(String customerName, int accountNumber, String accountType, double balance) {
        super(customerName, accountNumber, accountType, balance);
    }

    void deposit(double amount) {
        balance += amount;
        System.out.println("Amount deposited: " + amount);
    }

    void displayBalance() {
        System.out.println("Balance: " + balance);
    }
    void computeInterest() {
        double interestRate = 0.05;
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest added: " + interest);
    }

    void withdraw(double amount) {
        if (balance < amount) {
            System.out.println("Insufficient balance");
        } else {
            balance -= amount;
            System.out.println("Amount withdrawn: " + amount);
        }
    }
}

class CurrentAccount extends Account {
    double minimumBalance = 1000;
    double serviceCharge = 50;

    CurrentAccount(String customerName, int accountNumber, String accountType, double balance) {
        super(customerName, accountNumber, accountType, balance);
    }

    void deposit(double amount) {
        balance += amount;
        System.out.println("Amount deposited: " + amount);
    }

    void displayBalance() {
        System.out.println("Balance: " + balance);
    }
    void computeInterest() {
        System.out.println("Current account does not earn interest");
    }

    void withdraw(double amount) {
        if (balance - amount < minimumBalance) {
            System.out.println("Insufficient balance");
            balance -= serviceCharge;
            System.out.println("Service charge: " + serviceCharge);
        } else {
            balance -= amount;
            System.out.println("Amount withdrawn: " + amount);
        }
    }
}

class Brun {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter customer name: ");
        String customerName = sc.nextLine();

        System.out.print("Enter account number: ");
        int accountNumber = sc.nextInt();

        System.out.print("Enter account type (savings/current): ");
        String accountType = sc.next();

        System.out.print("Enter initial balance: ");
        double balance = sc.nextDouble();
```

```
System.out.println("\n");

Account account;
if (accountType.equals("savings")) {
   account = new SavingsAccount(customerName, accountNumber, accountType, balance);
} else {
   account = new CurrentAccount(customerName, accountNumber, accountType, balance);
}

while (true) {
   System.out.println("\n1. Deposit");
   System.out.println("2. Display balance");
   System.out.println("3. Compute interest");
   System.out.println("4. Withdraw");
   System.out.println("5. Exit\n");

   System.out.print("Enter choice: ");
   int choice = sc.nextInt();

   switch (choice) {
      case 1:
         System.out.print("\nEnter amount to deposit: ");
         double amount = sc.nextDouble();
         account.deposit(amount);
         break;
      case 2:
         account.displayBalance();
         break;
      case 3:
         account.computeInterest();
         break;
      case 4:
         System.out.print("\nEnter amount to withdraw: ");
```

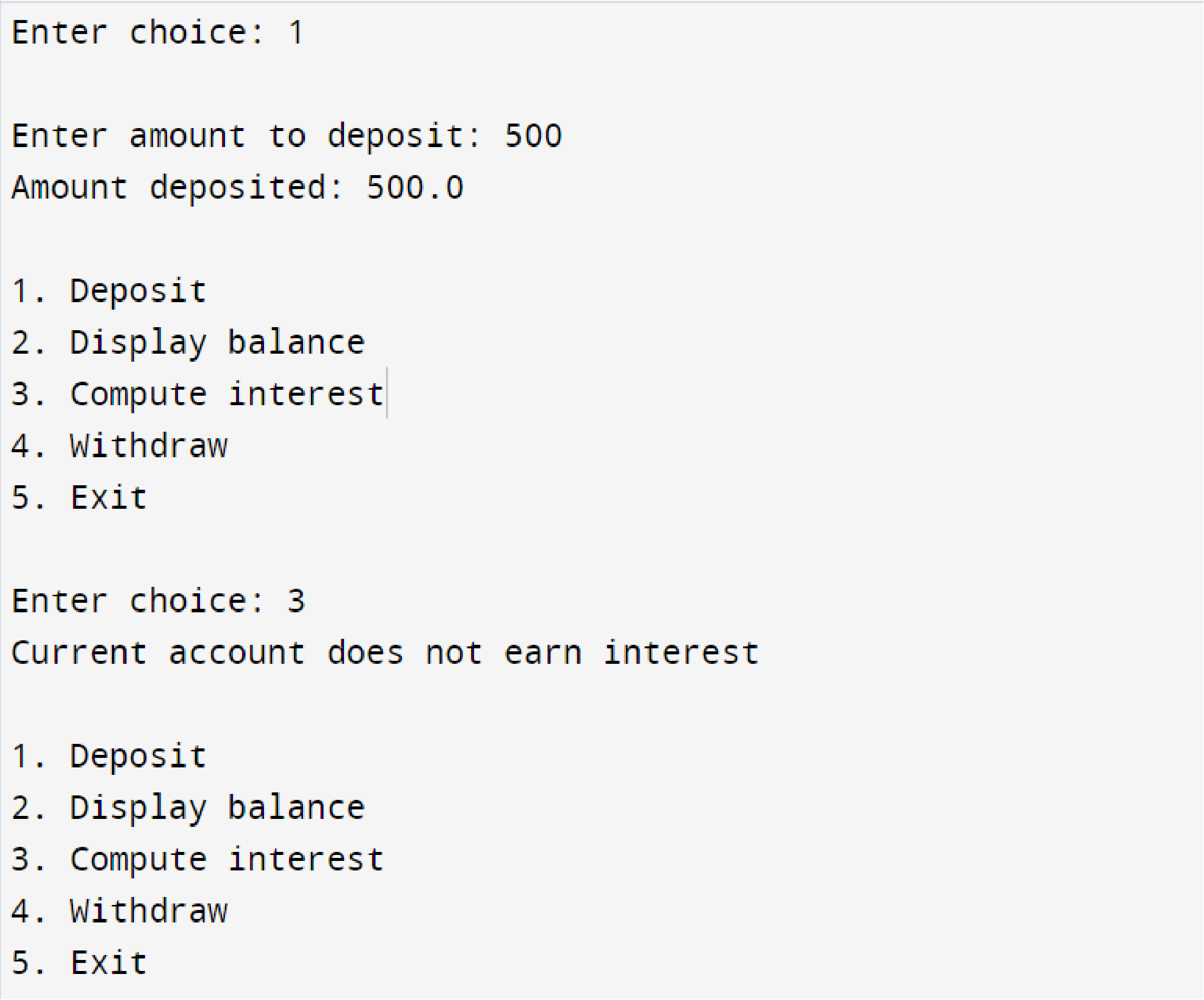
```
                amount = sc.nextDouble();
                account.withdraw(amount);
                break;
            case 5:
                sc.close();
                System.exit(0);
                break;
            default:
                System.out.println("\nInvalid choice");
        }
    }
  }
}
```

KEERTHI REDDY

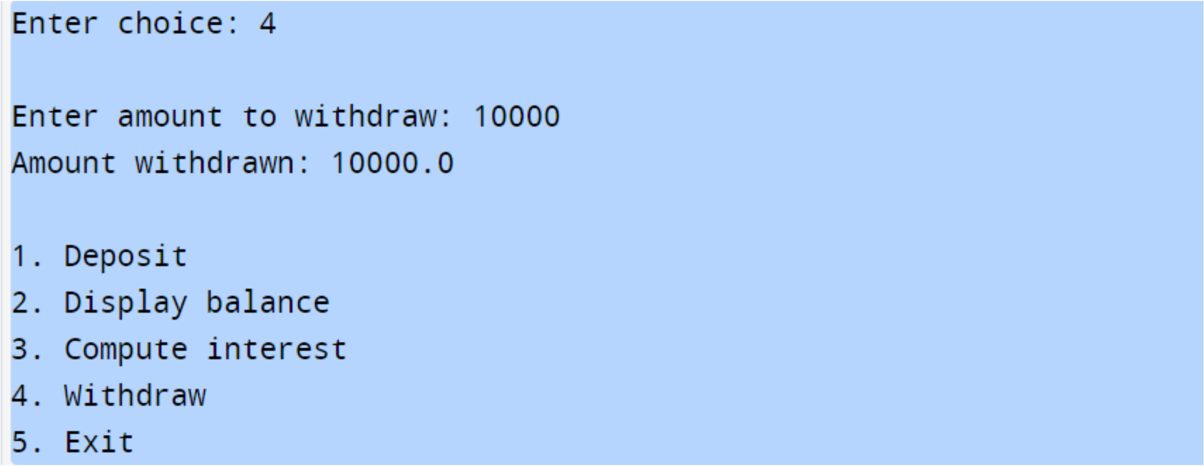
USN: 1BM22CS094



```
java -cp /tmp/pw1qEJZRkF Brun
Enter customer name: KEERTHI
Enter account number: 12345
Enter account type (savings/current): SAVINGS
Enter initial balance: 20000




1. Deposit
2. Display balance
3. Compute interest
4. Withdraw
5. Exit

Enter choice: 2
Balance: 20000.0

1. Deposit
2. Display balance
3. Compute interest
```



```
Enter choice: 1

Enter amount to deposit: 500
Amount deposited: 500.0

1. Deposit
2. Display balance
3. Compute interest
4. Withdraw
5. Exit

Enter choice: 3
Current account does not earn interest

1. Deposit
2. Display balance
3. Compute interest
4. Withdraw
5. Exit
```



```
Enter choice: 4

Enter amount to withdraw: 10000
Amount withdrawn: 10000.0

1. Deposit
2. Display balance
3. Compute interest
4. Withdraw
5. Exit
```

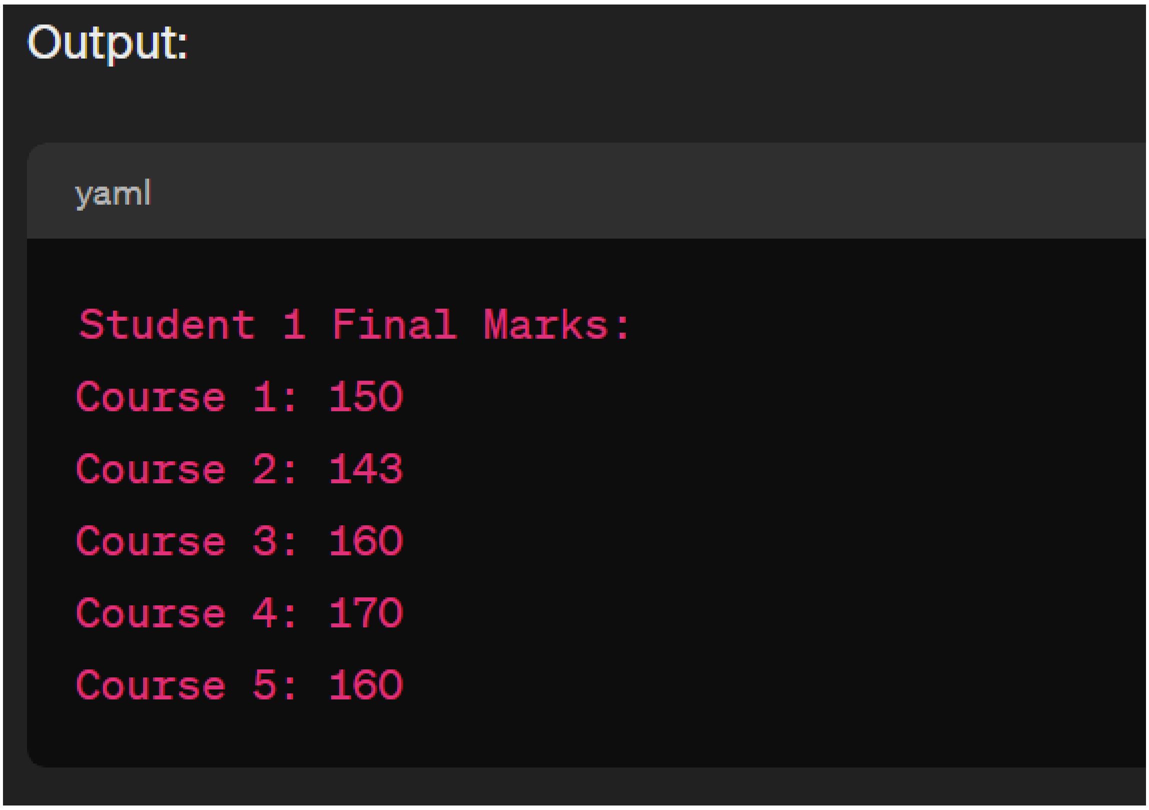
4) STUDENTS MARKS

```
// File: CIE/Student.java

package CIE;


public class Student {
    protected String usn;
    protected String name;
    protected int sem;

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



// File: CIE/Internals.java
package CIE;

public class Internals extends Student {
    protected int[] internalMarks = new int[5];

    public Internals(String usn, String name, int sem, int[] internalMarks) {
        super(usn, name, sem);
        this.internalMarks = internalMarks;
    }
}

// File: SEE/External.java
package SEE;
import CIE.*;

public class External extends Student {
    protected int[] externalMarks = new int[5];

    public External(String usn, String name, int sem, int[] externalMarks) {
        super(usn, name, sem);
        this.externalMarks = externalMarks;
    }
}

// File: Main.java
import CIE.*;
import SEE.*;

public class Main {
    public static void main(String[] args) {
        // Example usage
        // Internal marks for student 1
        int[] internalMarks1 = {80, 75, 85, 90, 88};
        Internals student1Internal = new Internals("1MS16CS001", "Alice", 3, internalMarks1);

        // External marks for student 1
        int[] externalMarks1 = {70, 68, 75, 80, 72};
        External student1External = new External("1MS16CS001", "Alice", 3, externalMarks1);

        // Display final marks for student 1
        System.out.println("Student 1 Final Marks:");
        for (int i = 0; i < 5; i++) {
            int finalMarks = student1Internal.internalMarks[i] + student1External.externalMarks[i];
            System.out.println("Course " + (i + 1) + ": " + finalMarks);
        }
    }
}
```

KEERTHI REDDY

USN: 1BM22CS094

OUTPUT:



Output:

yaml

```
Student 1 Final Marks:
Course 1: 150
Course 2: 143
Course 3: 160
Course 4: 170
Course 5: 160
```

6) EXCEPTIONAL HANDLING

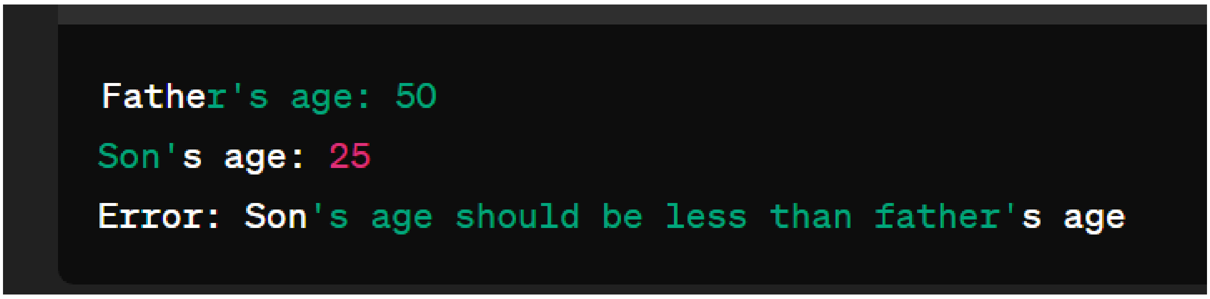
```
import java.util.Scanner;

class WrongAge extends Exception {
    public WrongAge() {
        super("Invalid age!");
    }
}

class Father {
    private int age;

    public Father(int age) throws WrongAge {
        if (age < 0) {
            throw new WrongAge();
        }
        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 >= fatherAge) {
            throw new WrongAge();
        }
        this.sonAge = sonAge;
    }

    public int getSonAge() {
        return sonAge;
    }
}

public class EMain{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        try {
            System.out.print("Enter father's age: ");
            int fatherAge = scanner.nextInt();

            System.out.print("Enter son's age: ");
```

```
      int sonAge = scanner.nextInt();

      Father father = new Father(fatherAge);
      System.out.println("Father's age: " + father.getAge());

      Son son = new Son(fatherAge, sonAge);
      System.out.println("Son's age: " + son.getSonAge());
    } catch (WrongAge e) {
      System.out.println(e.getMessage());
    } catch (Exception e) {
      System.out.println("Invalid input.");
    } finally {
      scanner.close();
    }
  }
```

KEERTHI REDDY

USN: 1BM22CS094

OUTPUT:



```
Father's age: 50
Son's age: 25
Error: Son's age should be less than father's age
```

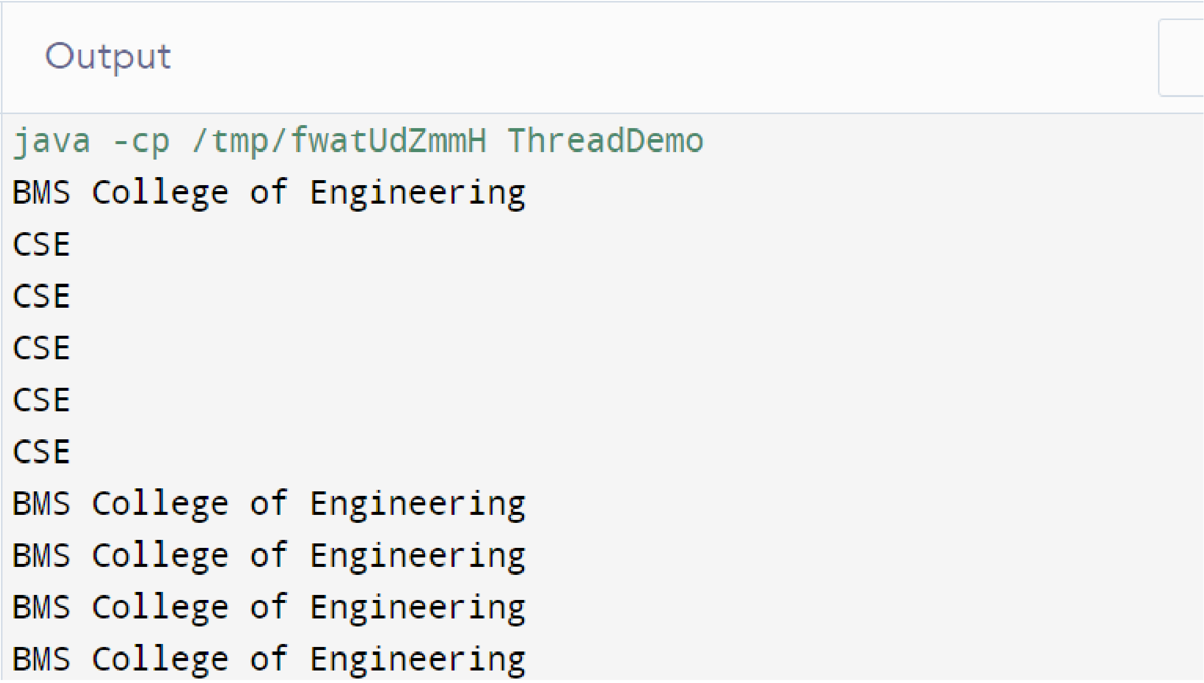
8) MULTI-THREADING

```
class DisplayThread extends Thread {
  private String message;
  private int interval;
```

```
    public DisplayThread(String message, int interval) {
        this.message = message;
        this.interval = interval;
    }

    public void run() {
        try {
            for(int i = 0; i < 5; i++) {
                System.out.println(message);
                Thread.sleep(interval * 1000);
            }
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

class ThreadDemo {
    public static void main(String[] args) {
        DisplayThread thread1 = new DisplayThread("BMS College of Engineering", 10);
        thread1.start();

        DisplayThread thread2 = new DisplayThread("CSE", 2);
        thread2.start();
    }
}
```

KEERTHI REDDY

USN:1BM22CS094



```
java -cp /tmp/fwatUdZmmH ThreadDemo
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
```

THANK YOU