Status	Finished
Started	Saturday, 5 October 2024, 11:41 PM
Completed	Sunday, 6 October 2024, 12:19 AM
Duration	37 mins 11 secs

Question **1**

Correct

Marked out of 5.00

Create a class known as "BankAccount" with methods called deposit() and withdraw().

Create a subclass called SavingsAccount that overrides the withdraw() method to prevent withdrawals if the account balance falls below one hundred.

For example:

Result
Create a Bank Account object (A/c No. BA1234) with initial balance of $500: Deposit $1000 into account BA1234: New balance after depositing $1000: $1500.0 Withdraw $600 from account BA1234: New balance after withdrawing $600: $900.0 Create a SavingsAccount object (A/c No. SA1000) with initial balance of $300: Try to withdraw $250 from SA1000! Minimum balance of $100 required! Balance after trying to withdraw $250: $300.0

Answer: (penalty regime: 0 %)

Reset answer

```
class BankAccount {
    // Private field to store the account number
    private String accountNumber;

    // Private field to store the balance
    private double balance;

    // Constructor to initialize account number and balance
    public BankAccount(String accountNumber, double balance) {
        this.accountNumber = accountNumber;
        this.balance = balance;
    }

    // Method to deposit an amount into the account
    public void deposit(double amount) {
        // Increase the balance by the deposit amount
        balance += amount;
    }

    // Method to withdraw an amount from the account
    public void withdraw(double amount) {
        // Check if the balance is sufficient for the withdrawal
        if (balance >= amount) {
            // Decrease the balance by the withdrawal amount
            balance -= amount;
        } else {
            // Print a message if the balance is insufficient
            System.out.println("Insufficient balance");
        }
    }

    // Method to get the current balance
    public double getBalance() {
        // Return the current balance
        return balance;
    }
}

class SavingsAccount extends BankAccount {
    // Constructor to initialize account number and balance
    public SavingsAccount(String accountNumber, double balance) {
```

```
        // Call the parent class constructor
        super(accountNumber, balance);
    }

    // Override the withdraw method from the parent class
    @Override
    public void withdraw(double amount) {
        // Check if the withdrawal would cause the balance to drop below $100
        if (getBalance() - amount < 100) {
            // Print a message if the minimum balance requirement is not met
            System.out.println("Minimum balance of $100 required!");
```

	Expected	Got	
✓	Create a Bank Account object (A/c No. BA1234) with initial balance of $500: Deposit $1000 into account BA1234: New balance after depositing $1000: $1500.0 Withdraw $600 from account BA1234: New balance after withdrawing $600: $900.0 Create a SavingsAccount object (A/c No. SA1000) with initial balance of $300: Try to withdraw $250 from SA1000! Minimum balance of $100 required! Balance after trying to withdraw $250: $300.0	Create a Bank Account object (A/c No. BA1234) with initial balance of $500: Deposit $1000 into account BA1234: New balance after depositing $1000: $1500.0 Withdraw $600 from account BA1234: New balance after withdrawing $600: $900.0 Create a SavingsAccount object (A/c No. SA1000) with initial balance of $300: Try to withdraw $250 from SA1000! Minimum balance of $100 required! Balance after trying to withdraw $250: $300.0	✓

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

Create a class Mobile with constructor and a method basicMobile().

Create a subclass CameraMobile which extends Mobile class , with constructor and a method newFeature().

Create a subclass AndroidMobile which extends CameraMobile, with constructor and a method androidMobile().

display the details of the Android Mobile class by creating the instance. .

class Mobile{

}

class CameraMobile extends Mobile {

}

class AndroidMobile extends CameraMobile {

}

expected output:

Basic Mobile is Manufactured
Camera Mobile is Manufactured
Android Mobile is Manufactured
Camera Mobile with 5MG px
Touch Screen Mobile is Manufactured

For example:

Result
Basic Mobile is Manufactured Camera Mobile is Manufactured Android Mobile is Manufactured Camera Mobile with 5MG px Touch Screen Mobile is Manufactured

Answer: (penalty regime: 0 %)

```
// Base Mobile class
class Mobile {

    // Constructor
    public Mobile() {
        System.out.println("Basic Mobile is Manufactured");
    }

    // Method for basic mobile feature
    public void basicMobile() {
        System.out.println("Basic Mobile functionality");
    }
}

// Subclass CameraMobile that extends Mobile
class CameraMobile extends Mobile {

    // Constructor
    public CameraMobile() {
        super(); // Call to the parent class (Mobile) constructor
        System.out.println("Camera Mobile is Manufactured");
    }

    // Method for Camera Mobile feature
    public void newFeature() {
        System.out.println("Camera Mobile with 5MG px");
    }
```

```
}

// Subclass AndroidMobile that extends CameraMobile
class AndroidMobile extends CameraMobile {

    // Constructor
    public AndroidMobile() {
        super(); // Call to the parent class (CameraMobile) constructor
        System.out.println("Android Mobile is Manufactured");
    }

    // Method for Android Mobile feature
    public void androidMobile() {
        System.out.println("Touch Screen Mobile is Manufactured");
    }
}

// Main class to demonstrate the functionality
public class Main {
    public static void main(String[] args) {
        // Create an instance of AndroidMobile
        AndroidMobile android = new AndroidMobile();

        // Display the features of AndroidMobile
        android.newFeature(); // Camera feature
```

	Expected	Got	
✓	Basic Mobile is Manufactured Camera Mobile is Manufactured Android Mobile is Manufactured Camera Mobile with 5MG px Touch Screen Mobile is Manufactured	Basic Mobile is Manufactured Camera Mobile is Manufactured Android Mobile is Manufactured Camera Mobile with 5MG px Touch Screen Mobile is Manufactured	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

create a class called College with attribute String name, constructor to initialize the name attribute , a method called Admitted(). Create a subclass called CSE that extends Student class, with department attribute , Course() method to sub class. Print the details of the Student.

College:

String collegeName;

public College() { }

public admitted() { }

Student:

String studentName;

String department;

public Student(String collegeName, String studentName,String depart) { }

public toString()

Expected Output:

A student admitted in REC
CollegeName : REC
StudentName : Venkatesh
Department : CSE

For example:

Result
A student admitted in REC CollegeName : REC StudentName : Venkatesh Department : CSE

Answer: (penalty regime: 0 %)

Reset answer

```
// Base College class
class College {
    String collegeName;

    // Constructor to initialize college name
    public College(String collegeName) {
        this.collegeName = collegeName;
    }

    // Method to print admission message
    public void admitted() {
        System.out.println("A student admitted in " + collegeName);
    }
}

// Base Student class
class Student extends College {
    String studentName;
    String department;

    // Constructor
    public Student(String collegeName, String studentName, String department) {
        super(collegeName); // Call to the parent class constructor
        this.studentName = studentName;
        this.department = department;
    }
```

```
    // toString method to display student details
    @Override
    public String toString() {
        return "CollegeName : " + collegeName + "\n" +
               "StudentName : " + studentName + "\n" +
               "Department : " + department;
    }
}

// Subclass CSE that extends Student
class CSE extends Student {
    // Constructor
    public CSE(String collegeName, String studentName) {
        super(collegeName, studentName, "CSE"); // Call to the parent class constructor
    }

    // Course method can be added if needed
    public void course() {
        // Placeholder for additional functionality related to CSE course
    }
}

// Main class to demonstrate the functionality
public class Main {
    public static void main(String[] args) {
```

	Expected	Got	
✓	A student admitted in REC CollegeName : REC StudentName : Venkatesh Department : CSE	A student admitted in REC CollegeName : REC StudentName : Venkatesh Department : CSE	✓

Passed all tests! ✓

◄ Lab-05-MCQ

Jump to...

Is Palindrome Number? ►