CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show one page at a time

Finish review

Status	Finished
Started	Monday, 30 September 2024, 9:44 PM
Completed	Monday, 30 September 2024, 10:34 PM
Duration	50 mins 19 secs

Question **1**

Correct

Marked out of 5.00

Flag question

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 %)

```java
class Mobile {
    public Mobile() {
        System.out.println("Basic Mobile is Manufactured");
    }
}

class CameraMobile extends Mobile {
    public CameraMobile() {
        System.out.println("Camera Mobile is Manufactured");
    }

    public void newFeature() {
        System.out.println("Camera Mobile with 5MG px");
    }
}

class AndroidMobile extends CameraMobile {
    public AndroidMobile() {
        System.out.println("Android Mobile is Manufactured");
    }

    public void AndroidMobile() {
        System.out.println("Touch Screen Mobile is Manufactured");
    }
}

public class Main {
    public static void main(String[] args) {
        AndroidMobile androidMobile = new AndroidMobile();
        androidMobile.newFeature();
        androidMobile.AndroidMobile();
    }
}

```

	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 **2**

Correct

Marked out of 5.00

Flag question

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

```java
class College {
    protected String collegeName;

    public College(String collegeName) {
        // Initialize the instance variable
        this.collegeName = collegeName;
    }

    public void admitted() {
        System.out.println("A student admitted in " + collegeName);
    }
}

class Student extends College {
    String studentName;
    String department;

    public Student(String collegeName, String studentName, String department) {
        // Initialize the instance variables
        super(collegeName);
        this.studentName = studentName;
        this.department = department;
    }

    @Override
    public String toString() {
        // Return the details of the student
        return "CollegeName : " + collegeName + "\nStudentName : " + studentName + "\nDepartment : " + de
    }
}

public class Main {
    public static void main(String[] args) {
        // Create a new Student object
        Student s1 = new Student("REC", "Venkatesh", "CSE");

        // Invoke the admitted() method
        s1.admitted();

        // Print the details of the student
        System.out.println(s1.toString());
    }
}

```

	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! ✓

Question **3**

Incorrect

Marked out of 5.00

Flag question

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

```
// Base class for Bank Account
class BankAccount {
    protected String accountNumber;
    protected double balance;

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

    // Method to deposit money into the account
    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposit $" + amount + " into account " + accountNumber + ":");
            System.out.println("New balance after depositing $" + amount + ": $" + (int)balance); // Ca
        } else {
            System.out.println("Deposit amount must be positive!");
        }
    }

    // Method to withdraw money from the account
    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            System.out.println("Withdraw $" + amount + " from account " + accountNumber + ":"); // Prin
            balance -= amount;
            System.out.println("New balance after withdrawing $" + amount + ": $" + (int)balance); // C
        } else {
            System.out.println("Insufficient balance for withdrawal!");
        }
    }
}

// Subclass for Savings Account
class SavingsAccount extends BankAccount {
    private static final double MINIMUM_BALANCE = 100.0;

    // Constructor for SavingsAccount
    public SavingsAccount(String accountNumber, double initialBalance) {
        super(accountNumber, initialBalance);
    }

    // Overridden method to enforce minimum balance
    @Override
    public void withdraw(double amount) {
        System.out.println("Try to withdraw $" + amount + " from SA1000!");
        if (amount > 0 && (balance - amount) >= MINIMUM_BALANCE) {
            balance -= amount;
            System.out.println("New balance after withdrawing $" + amount + ": $" + (int)balance); // C
        } else {

```

	Expected	Got

```
✗ Create a Bank Account object (A/c No. BA1234) with initial balance of $500:   Create a Bank Account object (A/c
  Deposit $1000 into account BA1234:                                             Deposit $1000.0 into account BA12
  New balance after depositing $1000: $1500.0                                    New balance after depositing $100
  Withdraw $600 from account BA1234:                                             Withdraw $600.0 from account BA12
  New balance after withdrawing $600: $900.0                                     New balance after withdrawing $60
  Create a SavingsAccount object (A/c No. SA1000) with initial balance of $300:  Create a Savings Account object (
  Try to withdraw $250 from SA1000!                                              Try to withdraw $250.0 from SA100
  Minimum balance of $100 required!                                              Minimum balance of $100 required!
  Balance after trying to withdraw $250: $300.0                                  Balance after trying to withdraw
```

Your code must pass all tests to earn any marks. Try again.

Show differences

Finish review

← Lab-05-MCQ

Jump to... ⬍

Is Palindrome Number? →