Status	Finished
Started	Monday, 7 October 2024, 6:05 PM
Completed	Monday, 7 October 2024, 6:36 PM
Duration	30 mins 58 secs

Question **1**

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

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

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

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

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

    // Constructor to initialize the attributes
    public Student(String collegeName, String studentName, String department) {
        super(collegeName); // Call to the College 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;
    }
}
```

```
36
37 // CSE class that extends Student
38 class CSE extends Student {
39     // Constructor to initialize the attributes
40     public CSE(String collegeName, String studentName) {
41         super(collegeName, studentName, "CSE"); // Set department to CSE
42     }
43
44     // Method for course (optional)
45     public void course() {
46         System.out.println("Courses available in CSE.");
47     }
48 }
49
50 // Main class to execute the code
51 public class Main {
52     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! ✓

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 %)

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

    public void basicMobile() {
        System.out.println("Basic Mobile features.");
    }
}
class CameraMobile extends Mobile {
    public CameraMobile() {
        super();
        System.out.println("Camera Mobile is Manufactured");
    }
    public void newFeature() {
        System.out.println("Camera Mobile with 5MG px");
    }
}
class AndroidMobile extends CameraMobile {
    public AndroidMobile() {
        super();
        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(); // Call method from CameraMobile
        androidMobile.androidMobile(); // Call method from 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 **3**

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 String account;
    private double balance;
    BankAccount(String account,double balance){
        this.account=account;
        this.balance=balance;
    }
    public void deposit(double amount) {
        this.balance+=amount;
    }
    public void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
        } else {
            System.out.println("Insufficient balance");
        }
    }
    public double getBalance() {
        return this.balance;
    }
}
 class SavingsAccount extends BankAccount {
    public SavingsAccount(String account, double balance) {
        super(account,balance);
    }
    @Override
    public void withdraw(double amount) {
        if (getBalance() - amount < 100) {
            System.out.println("Minimum balance of $100 required!");
        } else {
            super.withdraw(amount);
        }
    }
}

class prog{
    public static void main(String[] args) {
        System.out.println("Create a Bank Account object (A/c No. BA1234) with initial balance of $500:");
        BankAccount BA1234 = new BankAccount("BA1234", 500);
        System.out.println("Deposit $1000 into account BA1234:");
        BA1234.deposit(1000);
        System.out.println("New balance after depositing $1000: $"+ BA1234.getBalance());
        System.out.println("Withdraw $600 from account BA1234:");
        BA1234.withdraw(600);
        System.out.println("New balance after withdrawing $600: $" + BA1234.getBalance());
        System.out.println("Create a SavingsAccount object (A/c No. SA1000) with initial balance of $300:");
        SavingsAccount SA1000 = new SavingsAccount("SA1000", 300);
        System.out.println("Try to withdraw $250 from SA1000!");
        SA1000.withdraw(250);
        System.out.println("Balance after trying to withdraw $250: $" + SA1000.getBalance());
```

```
51      }
52  }
```

	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! ✓

◄ Lab-05-MCQ

Jump to...

Is Palindrome Number? ►