Status	Finished
Started	Tuesday, 1 October 2024, 8:04 PM
Completed	Tuesday, 1 October 2024, 8:46 PM
Duration	42 mins 22 secs

Question **1**

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() {
    }
}

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

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 accountNumber;
    protected double balance;

    public BankAccount(String accountNumber, double initialBalance) {
        this.accountNumber = accountNumber;
        this.balance = initialBalance;
    }

    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit $" + (int) amount + " into account " + accountNumber + ":");
        System.out.println("New balance after depositing $" + (int) amount + ": $" + balance);
    }

    public void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdraw $" + (int) amount + " from account " + accountNumber + ":");
            System.out.println("New balance after withdrawing $" + (int) amount + ": $" + balance);
        } else {
            System.out.println("Insufficient balance");
        }
    }

    public String getAccountNumber() {
        return accountNumber;
    }
}

class SavingsAccount extends BankAccount {
    public SavingsAccount(String accountNumber, double initialBalance) {
        super(accountNumber, initialBalance);
    }

    @Override
    public void withdraw(double amount) {
        if (balance - amount < 100) {
            System.out.println("Try to withdraw $" + (int) amount + " from " + getAccountNumber() + "!");
            System.out.println("Minimum balance of $100 required!");
        } else {
```

```
            super.withdraw(amount);
        }
    }
}

public class Main {
    public static void main(String[] args) {
        BankAccount bankAccount = new BankAccount("BA1234", 500);
        System.out.println("Create a Bank Account object (A/c No. " + bankAccount.getAccountNumber() + ") with in
        bankAccount.deposit(1000);
```

	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 **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

```
class College {
    String collegeName;

    public College(String collegeName) {
        this.collegeName = collegeName;
    }

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

class Student {
    String studentName;
    String department;
    String collegeName;

    public Student(String collegeName, String studentName, String department) {
        this.collegeName = collegeName; // Initialize collegeName here
        this.studentName = studentName;
        this.department = department;
    }

    @Override
    public String toString() {
        return "CollegeName : " + collegeName + "\nStudentName : " + studentName + "\nDepartment : " + department
```

```
    }
}

class CSE extends Student {
    public CSE(String collegeName, String studentName) {
        super(collegeName, studentName, "CSE");
    }
}

public class Main {
    public static void main(String[] args) {
        College college = new College("REC");
        college.admitted();

        CSE student = new CSE(college.collegeName, "Venkatesh");
        System.out.println(student);
    }
}
```

	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? ►