Status	Finished
Started	Saturday, 5 October 2024, 7:43 PM
Completed	Saturday, 5 October 2024, 8:29 PM
Duration	45 mins 33 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 %)

```
import java.util.*;
 class Mobile{
    void basicMobile(){
        System.out.println("Basic Mobile is Ma
    }
}
 class CameraMobile extends Mobile{
    CameraMobile(){
        super.basicMobile();
        System.out.println("Camera Mobile is M
    }
    void newFeature(){
        System.out.println("Camera Mobile with
    }
}
 class AndroidMobile extends CameraMobile{
    AndroidMobile(){

        System.out.println("Android Mobile is
    }
    void androidMobile(){
        System.out.println("Touch Screen Mobil
    }
}
public class Main{
    public static void main(String[] args){
       AndroidMobile a=new AndroidMobile();
```

```
28          a.newFeature();
29          a.androidMobile();
30
31      }
32  }
```

	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 field to store the account numb
    private String accountNumber;

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

    BankAccount(String a,double b){
        accountNumber=a;
        balance=b;
    }




    // Method to deposit an amount into the ac
    public void deposit(double amount) {
        balance+=amount;

    }

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

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

    }
}

```

```
class SavingsAccount extends BankAccount {
    // Constructor to initialize account numbe
    public SavingsAccount(String accountNumber
        super(accountNumber,balance);


    }

    // Override the withdraw method from the p
    @Override
    public void withdraw(double amount) {
```

	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
{
protected   String collegeName;

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

public void admitted() {
    System.out.println("A student admitted in
}
}
class Student extends College{

String studentName;
String department;

public Student(String collegeName, String stud
   super(collegeName);
   this.studentName=studentName;
   department=depart;

}
public String n(){
    return collegeName;
}
public String toString(){
```

```
public String toString(){
   return studentName;

}
public String d(){
    return department;
}
}
public class Main {
public static void main (String[] args) {
        Student  s1 = new Student("REC","Venka
        s1.admitted();
        System.out.println("CollegeName : "+s1
        System.out.println("StudentName : "+s1
        System.out.println("Department : "+s1.
}
}
```

	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? ►