

[Dashboard](#) / [My courses](#) / [CS23333-OOPUI-2023](#) / [Lab-05-Inheritance](#) / [Lab-05-Logic Building](#)

| | |
|------------------|------------------------------------|
| Status | Finished |
| Started | Saturday, 5 October 2024, 11:26 PM |
| Completed | Saturday, 5 October 2024, 11:27 PM |
| Duration | 1 min 10 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 %)

```
1 class Mobile{  
2     public Mobile(){  
3  
4         System.out.println("Basic Mobile is Manufactured");  
5     }  
6 }  
7 class CameraMobile extends Mobile{  
8  
9     public CameraMobile(){  
10         System.out.println("Camera Mobile is Manufactured");  
11     }  
12     public void newFeature(){  
13         System.out.println("Camera Mobile with 5MG px");  
14     }  
15 }  
16  
17 class AndroidMobile extends CameraMobile{  
18     public AndroidMobile(){  
19         System.out.println("Android Mobile is Manufactured");  
20     }  
21     void androidMobile(){  
22         System.out.println("Touch Screen Mobile is Manufactured");  
23     }  
24 }  
25  
26 class prog{  
27     public static void main(String[] args){  
28         AndroidMobile o=new AndroidMobile();  
29         o.newFeature();  
30         o.androidMobile();  
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 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

```

1  class College
2  {
3      protected String collegeName;
4
5      public College(String collegeNameP) {
6          // initialize the instance variables
7          collegeName= collegeNameP;
8      }
9
10     public void admitted() {
11         System.out.println("A student admitted in "+collegeName);
12     }
13 }
14 class Student extends College{
15
16     String studentName;
17     String depart;
18
19     public Student(String collegeNameP, String studentNameP,String departP) {
20         // initialize the instance variables
21         super(collegeNameP);
22         studentName=studentNameP;
23         depart=departP;
24
25
26
27     }
28
29     public String toString(){
30         // return the details of the student
31         return "CollegeName : "+collegeName+"\nStudentName : "+studentName+"\nDepartment : "+depart ;
32     }
33 }
34 class prog {
35     public static void main (String[] args) {

```

```
36      Student s1 = new Student("REC", "Venkatesh", "CSE");
37
38      s1.admitted();                                // invoke the admitted() method
39      System.out.println(s1.toString());
40  }
41  }
```

| | 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

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

```
1 class BankAccount {
2     private String accountNumber;
3     private double balance;
4
5     public BankAccount(String accountNumber, double balance){
6         this.accountNumber=accountNumber;
7         this.balance=balance;
8     }
9
10    // Method to deposit an amount into the account
11    public void deposit(double amount) {
12        // Increase the balance by the deposit amount
13        balance+=amount;
14    }
15
16
17    public void withdraw(double amount) {
18        if (balance >= amount) {
19            balance -= amount;
20        } else {
21            System.out.println("Insufficient balance");
22        }
23    }
24
25    // Method to get the current balance
26    public double getBalance() {
27        // Return the current balance
28        return balance;
29    }
30 }
31
32
33 class SavingsAccount extends BankAccount {
34     // Constructor to initialize account number and balance
35     public SavingsAccount(String accountNumber, double balance) {
36         // Call the parent class constructor
37         super(accountNumber,balance);
38     }
39
40
41    // Override the withdraw method from the parent class
42    @Override
43    public void withdraw(double amount) {
44        // Check if the withdrawal would cause the balance to drop below $100
45        if (getBalance() - amount < 100) {
46            // Print a message if the minimum balance requirement is not met
47            System.out.println("Minimum balance of $100 required!");
48        } else {
49            // Call the parent class withdraw method
50            super.withdraw(amount);
51        }
52    }
53 }
```

```

51     }
52 }
```

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

Passed all tests! ✓

[◀ Lab-05-MCQ](#)

Jump to...

[Is Palindrome Number? ▶](#)