

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

Status	Finished
Started	Monday, 30 September 2024, 7:10 PM
Completed	Monday, 30 September 2024, 8:06 PM
Duration	56 mins 24 secs

Question 1

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 field to store the account number
3     private String accountNumber;
4
5     // Private field to store the balance
6     private double balance;
7
8     // Constructor to initialize account number and balance
9     public BankAccount(String accountNumber, double initialBalance) {
10         this.accountNumber = accountNumber;
11         this.balance = initialBalance;
12     }
13
14
15
16
17
18     // Method to deposit an amount into the account
19     public void deposit(double amount) {
20         // Increase the balance by the deposit amount
21         balance += amount;
22     }
23     // Method to withdraw an amount from the account
24     public void withdraw(double amount) {
25         // Check if the balance is sufficient for the withdrawal
26         if (balance >= amount) {
27             // Decrease the balance by the withdrawal amount
28             balance -= amount;
29         } else {
30             // Print a message if the balance is insufficient
31             System.out.println("Insufficient balance");
32         }
33     }
34
35     // Method to get the current balance
36     public double getBalance() {
37         // Return the current balance
38         return balance;
39     }
40 }
41
42 class SavingsAccount extends BankAccount {
43     // Constructor to initialize account number and balance
```

	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 **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 collegeName) {
6          // initialize the instance variables
7          this.collegeName = collegeName;
8      }
9
10     public void admitted() {
11         System.out.println("A student admitted in "+collegeName);
12     }
13     public String getCollegeName() {
14         return collegeName;
15     }
16 }
17 class Student extends College{
18
19     private String studentName;
20     private String department;
21
22     public Student(String collegeName, String studentName,String depart) {
23         // initialize the instance variables
24         super(collegeName); // Call the parent class constructor
25         this.studentName = studentName;
26         this.department = depart;
27     }
28 }
```

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

[◀ Lab-05-MCQ](#)

Jump to...

[Is Palindrome Number? ▶](#)