CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show one page at a time

Finish review

Status	Finished
Started	Saturday, 5 October 2024, 12:57 PM
Completed	Saturday, 5 October 2024, 1:14 PM
Duration	16 mins 42 secs

Question **1**
Correct
Marked out of 5.00
Flag question

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
class College {
    protected String collegeName;

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

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

class Student extends College {
    String studentName;
    String department;

    public Student(String collegeName, String studentName, String department) {
        super(collegeName);
        this.studentName = studentName;
        this.department = department;
    }

    public String toString() {
        return "CollegeName : " + collegeName + "\nStudentName : " + studentName + "\nDepartment : "
    }
}

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

public class Main {
    public static void main(String[] args) {
        CSE student = new CSE("REC", "Venkatesh");
        student.admitted();
        System.out.println(student.toString());
    }
}

```

Expected	Got
A student admitted in REC	A student admitted in REC
CollegeName : REC	CollegeName : REC
StudentName : Venkatesh	StudentName : Venkatesh
Department : CSE	Department : CSE

Passed all tests!

Question **2**
Correct
Marked out of 5.00
Flag question

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

```java
class BankAccount {
    private String accountNumber;
    private double balance;

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

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

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

    public double getBalance() {
        return balance;
    }
}

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

    @Override
    public void withdraw(int amount) {
        if (getBalance() - amount < 100) {
            System.out.println("Minimum balance of $100 required!");
        } else {
            super.withdraw(amount);
        }
    }
}

public class Main {
    public static void main(String[] args) {
        System.out.println("Create a Bank Account object (A/c No. BA1234) with initial balance of
        BankAccount BA1234 = new BankAccount("BA1234", 500);
        System.out.println("Deposit $1000 into account BA1234:");
        BA1234.deposit(1000);
        System.out.println("Withdraw $600 from account BA1234:");
        BA1234.withdraw(600);

```

Expected	Got
Create a Bank Account object (A/c No. BA1234) with initial balance of $500:	Create a Bank Account object (A/c No. B
Deposit $1000 into account BA1234:	Deposit $1000 into account BA1234:
New balance after depositing $1000: $1500.0	New balance after depositing $1000: $15
Withdraw $600 from account BA1234:	Withdraw $600 from account BA1234:
New balance after withdrawing $600: $900.0	New balance after withdrawing $600: $90
Create a SavingsAccount object (A/c No. SA1000) with initial balance of $300:	Create a SavingsAccount object (A/c No.
Try to withdraw $250 from SA1000!	Try to withdraw $250 from SA1000!
Minimum balance of $100 required!	Minimum balance of $100 required!
Balance after trying to withdraw $250: $300.0	Balance after trying to withdraw $250:

Passed all tests!

Question 3

Correct

Marked out of 5.00

Flag question

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 %)

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

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();
        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!

Finish review