Devanshu Surana
PC-23, 1032210755
Batch C1

OOCCJ Lab Assignment 3B

Problem statement: Write a Java program to create two Interfaces Motorbike and cycle. Motorbike interface ~~controls~~ consists of the attribute speed and the method totalDistance(). Cycle interface consists of the attribute Time. These interfaces are implemented by the class Twowheeler. Calculate totat distance travelled.

Objective:

1) To study abstract class.
2. To study interfaces in Java.

Theory:

1. Java Abstraction:

- A process of hiding the implementation details and showing only functionality to the user.

Shows only essential things to the user and hides the internal details.

Ex: sending SMS where you type the text and send the message. You don't know the internal processing about the message delivery.

It focuses on what the object does instead of how it does it.

There are 2 ways to achieve abstraction in Java:

1) Abstract class (0 to 100%)

2. Interface (100%)

2. Java Interfaces:
An Interface Is a completely "Abstract class" tha
used to group related methods with empty bod
By interface, we can support the functionality
Inheritance.
It can be used to achieve base coupling.
To access the interface methods, the interface
"implemented" by another class, with the imple
Keyword (instead of extends).

Conclusion:
Thus, we have successfully implemented interface

FAQ's

Ans 1) Abstract class cannot be in instantiated using new
while Concrete class can be instantiated using
Keyword.
Abstract class may or may not have abstract m
but concrete class can not have an abstract m
Abstract class cannot be declared as final class
Concrete class can be declared final.
Abstract class can have both an abstract a
concrede methods while A concrete class can
concrete methods. Even a single abstract meth
the Class abstract.

2) No, it is not possi
modifiers for the
the members defi
or in other word
in an interface

Algorithm:

step1: Start
step 2: Create Int
step.3: create c
two int
step 4: Calculat
Step 5: END.

2) No, it is not possible to define private and protected modifiers for the members in Interfaces in Java. As the members defined in interfaces are implicitly public or in other words, we can say the member defined in an interface is by default public.

Algorithm :

Step1 : Start
Step 2 : Create Interface motorbike and cycle.
Step. 3 : Create class two wheeler which implements the two interfaces.
Step 4 : Calculate total distance
Step 5 : END.

Name: Devanshu Surana

Roll No.: 23
Panel: C
Batch: C1

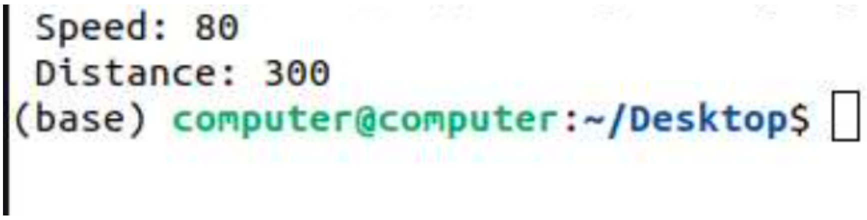
OOCCJ LAB ASSIGNMENT 3B

CODE:-

```
import java.io.*;

import java.lang.*;

import java.util.*;

interface Motorbike

{

  int speed = 80;

  public void totalDistance ();

}

interface Cycle

{

  int distance = 300;

  public void speed ();

}

class TwoWheeler implements Motorbike, Cycle

{

  Scanner myObj = new Scanner (System.in);

  public void totalDistance ()

  {                                        //System.out.println("Enter Speed");

//salary=myObj.nextInt(); System.out.println(" Speed: "+ speed);

  }
```

```
 public void speed ()
 {                                  //System.out.println("Enter Distance");
//salary=myObj.nextFloat(); System.out.println(" Distance: "+ distance); }
 }
 class devanshu
 {
  public static void main (String args[])
  {
   TwoWheeler Activa = new TwoWheeler ();
    Activa.totalDistance ();
    Activa.speed ();
 }


 }
```

OUTPUT



```
 Speed: 80
 Distance: 300
(base) computer@computer:~/Desktop$ 
```