B.Sc. Engg. SWE 3rd Semester (50) 20th June, 2021

# ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

## ORGANISATION OF ISLAMIC COOPERATION (OIC)

## Department of Computer Science and Engineering (CSE)

**MID SEMESTER EXAMINATION** **WINTER SEMESTER, 2020-2021**

**DURATION: 1 Hour 30 Minutes** **FULL MARKS: 75**

# SWE 4301: Object Oriented Concepts II

**Programmable calculators are not allowed.**

Figures in the right margin indicate marks and (COs, POs). You need to answer all three questions.
Write Student ID and Name top of the first page and write studentID and page no in every page of the answer script.
Submission pdf should be named as Full_Student_ID<space>Course_Code<dot>pdf

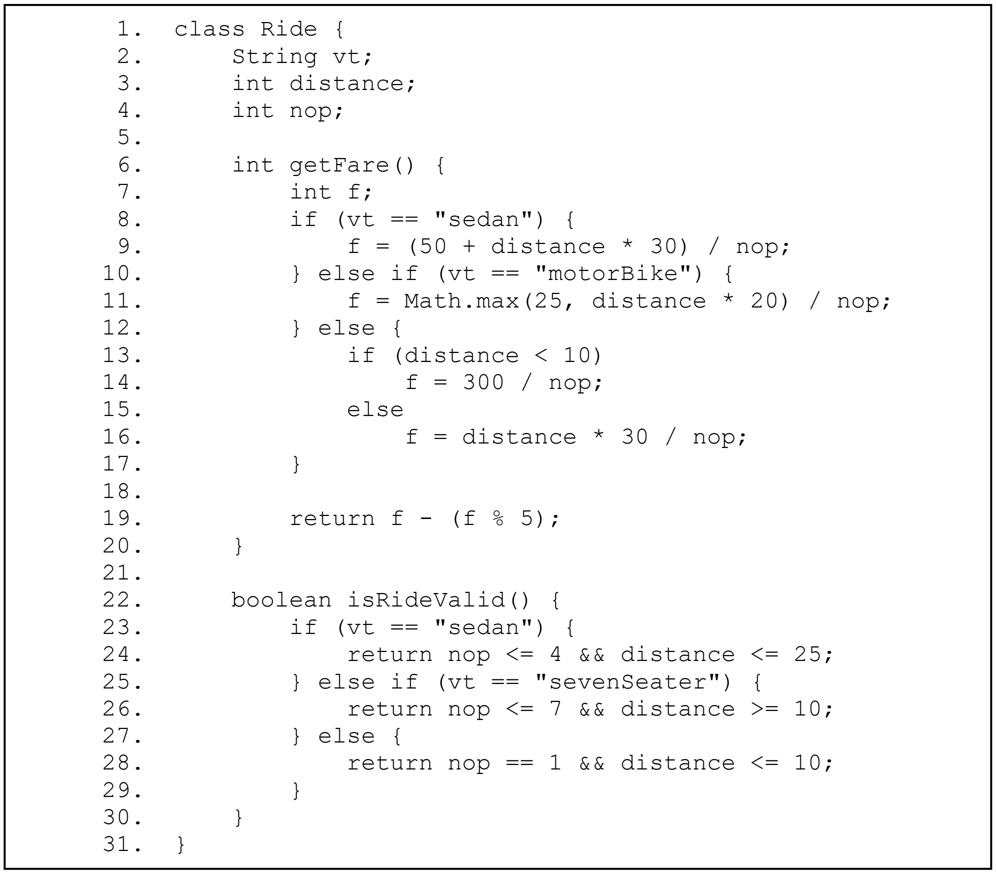
**1. a)** State the fundamental concept of the following design principles with examples. **12** **(CO3,PO3)**

i. Single-Responsibility Principle (SRP)
ii. Liskov Substitution Principle (LSP)
iii. Dependency Inversion Principle (DIP)

**b)** Both SRP and ISP deal with Cohesion. Describe how they are different. **6** **(CO3,PO3)**

**c)** Explain how code smell is removed by refactoring and refactoring requires Unit Test. Justify your answer. **7** **(CO3,PO3)**

**2. a)** A code snippet is given below. Your task is to find as many code-smells as possible (At least 5). Mention the line numbers where a code smell occurs. **10** **(CO2,PO2,PO4)**



```
 1.  class Ride {
 2.      String vt;
 3.      int distance;
 4.      int nop;
 5.
 6.      int getFare() {
 7.          int f;
 8.          if (vt == "sedan") {
 9.              f = (50 + distance * 30) / nop;
10.          } else if (vt == "motorBike") {
11.              f = Math.max(25, distance * 20) / nop;
12.          } else {
13.              if (distance < 10)
14.                  f = 300 / nop;
15.              else
16.                  f = distance * 30 / nop;
17.          }
18.
19.          return f - (f % 5);
20.      }
21.
22.      boolean isRideValid() {
23.          if (vt == "sedan") {
24.              return nop <= 4 && distance <= 25;
25.          } else if (vt == "sevenSeater") {
26.              return nop <= 7 && distance >= 10;
27.          } else {
28.              return nop == 1 && distance <= 10;
29.          }
30.      }
31.  }
```

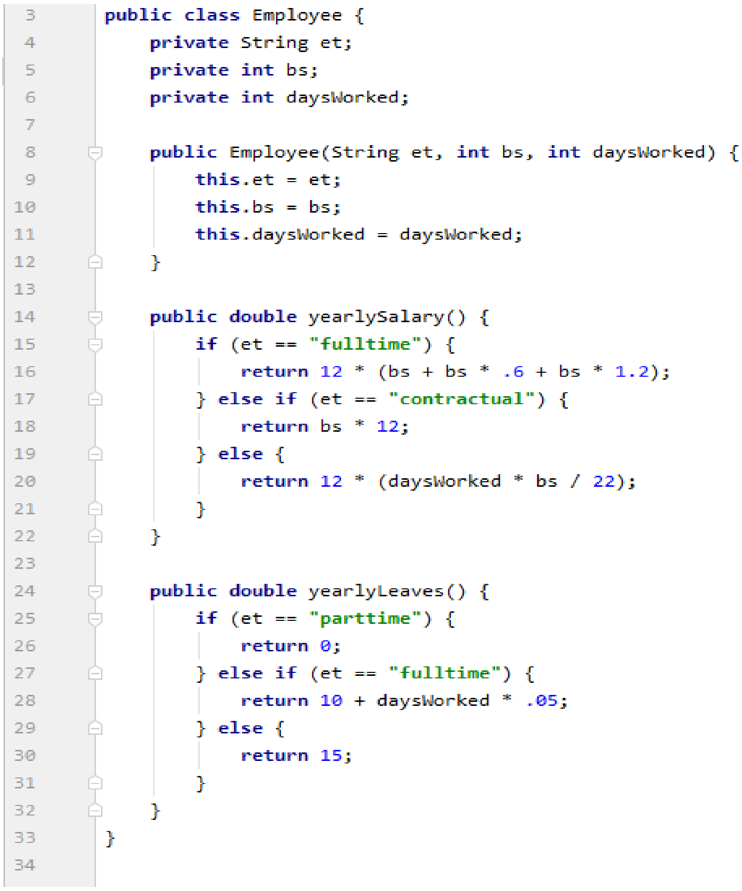
**Figure - 1: Ride Class**

**b)** Answer the following questions with respect to the code snippet in Figure – 1. **5+10** **(CO2,PO2,PO4)**

   i. Does the code in Figure - 1 violate any SOLID Principles? If yes, then what are those? Justify your answer.

   ii. The code in Figure – 1 needs refactoring. Remove the code smells and the violation of SOLID principles (If Any). Then re-write the expected code after refactoring.

**3.**



```java
public class Employee {
    private String et;
    private int bs;
    private int daysWorked;

    public Employee(String et, int bs, int daysWorked) {
        this.et = et;
        this.bs = bs;
        this.daysWorked = daysWorked;
    }

    public double yearlySalary() {
        if (et == "fulltime") {
            return 12 * (bs + bs * .6 + bs * 1.2);
        } else if (et == "contractual") {
            return bs * 12;
        } else {
            return 12 * (daysWorked * bs / 22);
        }
    }

    public double yearlyLeaves() {
        if (et == "parttime") {
            return 0;
        } else if (et == "fulltime") {
            return 10 + daysWorked * .05;
        } else {
            return 15;
        }
    }
}
```

**Figure – 2: Employee Class**

**a)** Describe the necessities of TDD. Write at least 3 (three) Unit Test methods for the code snippet/class in Figure - 2. **3+9** **(CO1,PO1,PO5)**

**b)** From each sets, select one class/method/variable name that you'll use in your code. **5** **(CO1,PO1,PO5)**

   i. (ETD, ElapsedTimeInDays)

   ii. (USB, UniversalSerialBus)

   iii. (GPS, GlobalPositioningSystem)

   iv. (genyymmdd, GenerationTimeStamp)

   v. (const white, const WHITE)

**c)** Explain why bad codes usually happen. Do you think codes should be easy to change? Justify your answer. **4+4** **(CO1,PO1,PO5)**