CONFIDENTIAL

UTM Faculty of Computing
UNIVERSITI TEKNOLOGI MALAYSIA

UNIVERSITI TEKNOLOGI MALAYSIA
TEST 2 (PRACTICAL)
SEMESTER II 2022/2023

SUBJECT CODE	**: SECJ2154**
SUBJECT NAME	**: OBJECT ORIENTED PROGRAMMING**
YEAR/COURSE	**: 1 (SECB/ SECJ/ SECP/ SECR/ SECV)**
TIME	**: 08:00 PM– 10:00 PM MYT (2 hours)**
DATE	**: 6th JUNE 2023 (Tuesday)**

INSTRUCTIONS TO THE STUDENTS:

- Read the problem and instructions carefully.
- References to any resources by any means except **OOP Lab Module** are strictly prohibited.
- You are given **TWO HOURS** to complete the test, including downloading test-related materials and submitting your answer.
- Your program must follow the input and output as required in the text and shown in the examples. You must test the programs with (but not limited to) all the input given in the examples.

IMPORTANT NOTES:

- All the **COMMENT STATEMENTS** in the submitted program **WILL NOT BE EVALUATED**.

SUBMISSION PROCEDURE:

- Only the source code (i.e. the file with the extension ***.java**) is required for the submission.
- You do not need to compress the file.
- Submit the source code file via the **UTM's e-learning system**.

This question book consists of **9 (NINE)** printed pages excluding this page.

QUESTION 1 – ERROR DEBUGGING [30 Marks]

You are given a Java program `TestShape3D.java` with errors (syntax and/ or logical errors). The program defines four classes that are the `TestShape3D`, `Shape3D`, `Cylinder`, and `Sphere`. Brief descriptions of each class and its methods are as follows:

Class Name	Class and Method Description
`TestShape3D`	• Provides the `main` method to run the program with a conditional loop to allow user to choose the 3D shape objects (cylinder or sphere) or to exit the program. • Based on 3D shape object choose by the users the program then calls the `createCylinder` or `createSphere` methods of the `Shape3D` class. An object of `Cylinder` or `Sphere` class returned by these methods is then stored into `ArrayList` object. • At the end, when the user chooses to exit, the program iterates through the items of the `ArrayList` object to list the 3D shape object type, its volume, and the total volume of all the shapes.
`Shape3D`	• Provides the `createCylinder` and `createSphere` methods to respectively create the cylinder or sphere 3D shape object. • The `createCylinder` method prompts the user to enter the radius and length of the cylinder. It then creates and returns a `Cylinder` object based on the given input. • The `createSphere` method prompts the user to enter the radius of the sphere. It then creates and returns a `Sphere` object based on the given input.
`Cylinder`	• Defines two instance variables, `radius` and `height`, to represent the properties of cylinder 3D shape object. • Provides constructor to received and set the values of the cylinder's properties. • Provides `getVolume` method to calculate and return the volume of the cylinder
`Sphere`	• Defines one instance variable, `radius`, to represent the property of sphere 3D shape object. • Provides constructor to received and set the values of the sphere's property • Provides `getVolume` method to calculate and return the volume of the sphere

Below (Figure 1.1) is the source code of the `TestShape3D.java` program. Open the program and write your name and matriculation number on lines 4 and 5 of the source code before proceeding to the next instructions.

```
01 // TestShape3D.java
02 // TEST 2 - Question 1
03 // SECJ2154 - 2022/2023-2
04 // Name: ???
05 // Matric No.: ???
06
07 import java.util.ArrayList;
08 import java.util.Scanner;
09
10 public class TestShape3D {
11   public static void Main(String[] args) {
12     Boolean exit = false;
13     String choice;
14     Scanner scan = new Scanner(System.in);
15
16     ArrayList objList = new ArrayList();
17     Shape3D s3d = new Shape3D();
18
19     System.out.println("Test Shape3D class");
20
21     while (exit) {
22       System.out.print("\nEnter your choice [cylinder | sphere |  exit ]:
23 ");
24       choice = scan.nextLine();
25
26       if (choice == "cylinder") {
27         objList.add(s3d.createCylinder(scan));
28
29       } else if (choice == "sphere") {
30         objList.add(s3d.createSphere(scan));
31
32       } else if (choice.equals("exit")) {
33         exit = true;
34       }
35     }
36
37     for (int i = 0; i < objList.size(); i++) {
38       if (objList.get(i) instanceof Cylinder) {
39         Cylinder obj = objList.get(i);
40         System.out.printf("Object #%d Type: Cylinder, Volume: %.3f\n",
41         (i + 1), obj.getVolume());
42       } else {
43         Cylinder obj = (Sphere) objList.get(i);
44         System.out.printf("Object #%d Type: Sphere, Volume: %.3f\n",
45         (i + 1), obj.getVolume());
46       }
47     }
48
49     System.out.print("TOTAL VOLUME = %.2f\n", Shape3D.TOTAL_VOLUME);
50   }
51 }
52
53 class Shape3D {
54   public final double PI = 3.14;
55
56   public double TOTAL_VOLUME = 0.0;
57
58   public void createCylinder(Scanner scn) {
59     System.out.println("Create Cylinder...");
60
```

```
    System.out.print("Radius: ");
    int r = scn.nextInt();

    System.out.print("Length: ");
    int l = scn.nextInt();

    // Remove \n from input buffer
    scn.nextLine();

    // Create new Cylinder instance and sum TOTAL_VOLUME
    Cylinder cyl = new Cylinder();
    TOTAL_VOLUME += cyl.getVolume();

    // Return the new created Cylinder instance
    return cyl;
  }

  public void createSphere(Scanner scn) {
    System.out.println("Create Sphere...");

    System.out.print("Radius: ");
    double r = scn.nextDouble();

    // Remove \n from input buffer
    scn.nextLine();

    // Create new Sphere instance and sum TOTAL_VOLUME
    Sphere sph = new Cylinder(l, w, h);
    TOTAL_VOLUME += sph.getVolume();

    // Return the new created Sphere instance
    return sph;
  }
}

class Cylinder {
  private double radius, height;

  // Constructor
  public Cylinder() { }

  // Constructor
  public Cylinder(int r, int h) {
    radius = r;
    length = h;
  }

  public double getVolume() {
    return Shape3D.PI * Math.pow(radius, 2) * height;
  }
}

class Sphere {
  private double radius;

  // Constructor
  public void Sphere(r) {
    radius = r;
  }

  public int getVolume() {
    return 4.0 / 3.0 * Shape3D.PI * Math.pow(radius, 3);
  }
}
```

Figure 1.1: The source code of `TestShape3D.java` program

Study how all of the classes were instantiated and used in the program (Figure 1.1) . Debug the errors, compile, and run the program. You are **NOT ALLOWED** to **remove** or **change** any statements that do not cause errors in the program. Only **update** statements that cause errors in the program.

Test your Java program by using the input test case (**`bold`** text) as shown in Figure 1.2 below:

```
Test Shape3D class

Enter your choice [cylinder | sphere |  exit ]: box

Enter your choice [cylinder | sphere |  exit ]: cylinder
Create Cylinder...
Radius: 2
Length: 3

Enter your choice [cylinder | sphere |  exit ]: sphere
Create Sphere...
Radius: 2.5

Enter your choice [cylinder | sphere |  exit ]: cylinder
Create Cylinder...
Radius: 2.5
Length: 2.2

Enter your choice [cylinder | sphere |  exit ]: exit
Object #1 Type: Cylinder, Volume: 37.680
Object #2 Type: Sphere, Volume: 65.417
Object #3 Type: Cylinder, Volume: 43.175
TOTAL VOLUME = 146.27
```

Figure 1.2: The input test case to run and test the `TestShape3D.java` program

QUESTION 2 – PROGRAMMING [70 Marks]

You have been assigned the task of developing the Sport Facility Management System, and required to complete the given source code - `SportClubFacilityBooking.java`. A class diagram, displayed in Figure 2.1 below, is provided to you. This diagram presents the definition and relationship of various classes, including `Person`, `ClubMember`, `ClubStaff`, `Address`, `Booking`, `SportClubFacility`, and public class `SportClubFacilityBooking`.

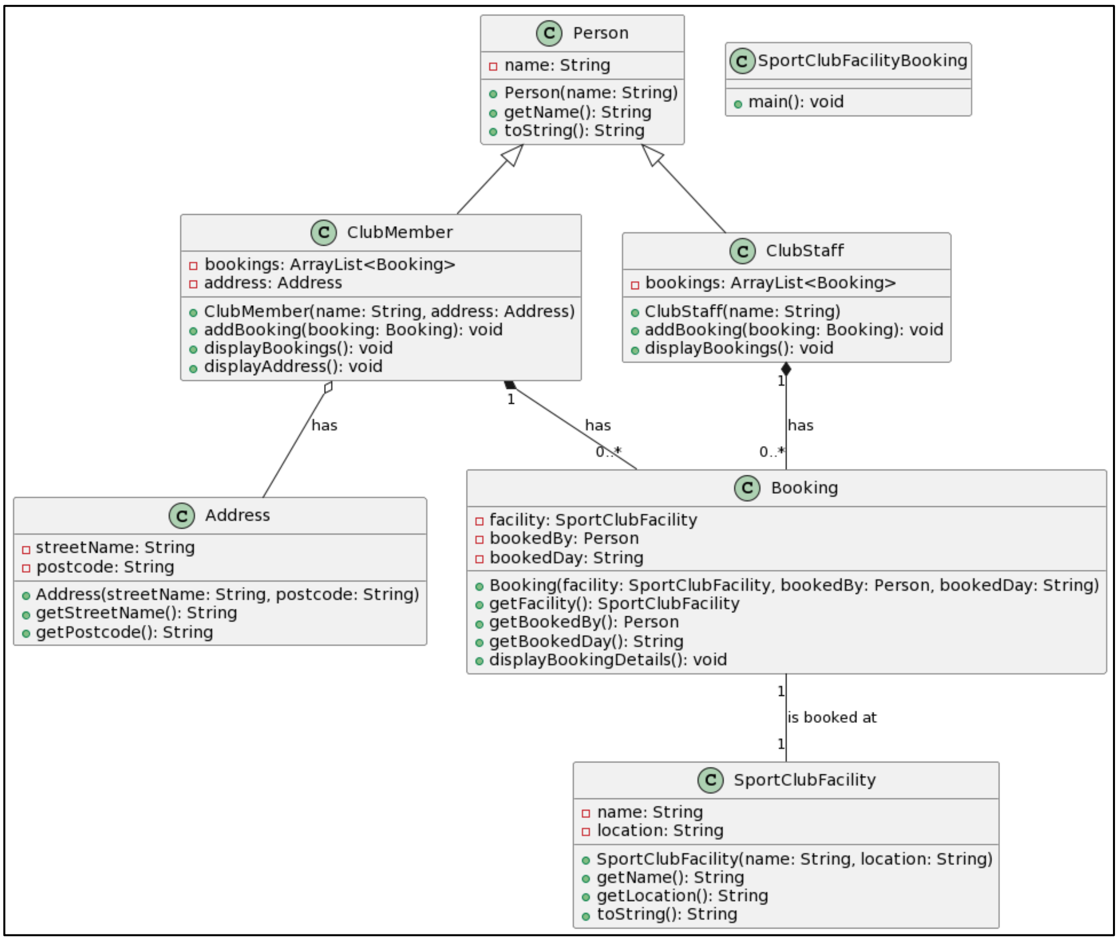

Figure 2.1: UML Class Diagram for Sport Booking Facility System

Figure 2.2 is the source code of the `SportClubFacilityBooking.java` program. Open the program and write your name and matriculation number on lines 4 and 5 of the source code before proceeding to the next instructions.

```
// SportClubFacilityBooking.java
// TEST 2 - Question 2
// SECJ2154 - 2022/2023-2
// Name: ???
// Matric No.: ???

import java.util.ArrayList;

class Person {
}

class Address {
}

class ClubMember extends Person {
}

class ClubStaff extends Person {
}

class SportClubFacility {
}

class Booking {
}

public class SportClubFacilityBooking {
  public static void main(String[] args) {
    System.out.println("\n\nLab Test - SportClub Facility Booking System\n");

    // checkpoint #1 - Create 2 sportclub facilities
    System.out.println("Checkpoint #1");
    // SportClubFacility facility1 = new SportClubFacility("Futsal Court",
    // "Taman U Sport Centre");
    // SportClubFacility facility2 = new SportClubFacility("Swimming Pool",
    // "Taman Tun Aquatic Center");
    // System.out.println(facility1.toString());
    // System.out.println(facility2.toString());

    // checkpoint #2 - Create 1 club member
    System.out.println("\nCheckpoint #2");
    // Address clubmemberAddress1 = new Address("No 45 Kolej Tun Dr Ismail",
    // "81300");
    // ClubMember clubMember1 = new ClubMember("Karim Marwari",
    // clubmemberAddress1);
    // System.out.println(clubMember1.toString());
    // clubMember1.displayAddress();
    // clubMember1.displayBookings();

    // checkpoint #3 - Create 1 club staff
    System.out.println("\nCheckpoint #3");
    // ClubStaff clubStaff1 = new ClubStaff("Famdari Akhazi");
    // System.out.println(clubStaff1.toString());
    // clubStaff1.displayBookings();

    // checkpoint #4 - Create 2 bookings
    System.out.println("\nCheckpoint #4");
    // Booking booking1 = new Booking(facility1, clubMember1,
    // "Sunday: 2023-06-11");
    // Booking booking2 = new Booking(facility2, clubStaff1,
    // "Thursday: 2023-06-15");
    // Booking booking3 = new Booking(facility1, clubStaff1,
    // "Friday: 2023-06-16");
    // booking1.displayBookingDetails();
    // booking2.displayBookingDetails();
    // booking3.displayBookingDetails();

    // checkpoint #5 - assign and display bookings by
    // club members and club staff
    System.out.println("\nCheckpoint #5");
    // clubMember1.addBooking(booking1);
    // clubStaff1.addBooking(booking2);
    // clubStaff1.addBooking(booking3);
    // clubMember1.displayBookings();
    // clubStaff1.displayBookings();
  }
}
```

Figure 2.2: An incomplete source code of `SportClubFacilityBooking.java` program

Open and run the `SportClubFacilityBooking.java` program. The initial output of the program should as shown below:

```
Lab Test - SportClub Facility Booking System

Checkpoint #1

Checkpoint #2

Checkpoint #3

Checkpoint #4

Checkpoint #5
```

Based on the class diagram and source code given in Figures 2.1 and 2.2, do the following tasks (i) -to- (v):

i. Complete the implementation of `SportClubFacility` class (implement the constructor, `getName`, `getLocation`, override `toString)`. Inside the `main` method of the `SportClubFacilityBooking` class, uncomment the checkpoint #1 so it can generate the output as shown below:

```
Checkpoint #1
SportClubFacility [name=Futsal Court, location=Taman U Sport Centre]
SportClubFacility [name=Swimming Pool, location=Taman Tun Aquatic Center]
```

(10 marks)

ii. Complete the implementation of the `Person`, `Address`, and `ClubMember` classes. Always refer to the class diagram in Figure 2.1 for all types of constructors and methods you need to implement in all of these classes. Uncomment the checkpoint #2 so it can generate the output as shown below:

```
Checkpoint #2
Person [name=Karim Marwari]
Address: No 45 Kolej Tun Dr Ismail, 81300
No booking made by Karim Marwari
```

(20 marks)

iii. Complete the implementation of `ClubStaff` class. Uncomment the checkpoint #3 so it can generate the output as shown below:

```
Checkpoint #3
Person [name=Famdari Akhazi]
No booking made by Famdari Akhazi
```

(15 marks)

iv. Complete the implementation of `Booking` class. Uncomment the checkpoint #4 so it can generate the output as shown below:

```
Checkpoint #4
Booking Details:
Facility: Futsal Court
Booked By: Karim Marwari
Booked Day: Sunday: 2023-06-11

Booking Details:
Facility: Swimming Pool
Booked By: Famdari Akhazi
Booked Day: Thursday: 2023-06-15

Booking Details:
Facility: Futsal Court
Booked By: Famdari Akhazi
Booked Day: Friday: 2023-06-16
```

(15 marks)

v. Verify that the `addBooking` and `displayBookings` methods inside the `ClubMember` and `ClubStaff` classes have been implemented correctly. Uncomment the checkpoint #5 so it can generate the output as shown below:

```
Checkpoint #5
Booking for member: Karim Marwari
Booking Details:
Facility: Futsal Court
Booked By: Karim Marwari
Booked Day: Sunday: 2023-06-11

Booking for staff:
Booking Details:
Facility: Swimming Pool
Booked By: Famdari Akhazi
Booked Day: Thursday: 2023-06-15

Booking Details:
Facility: Futsal Court
Booked By: Famdari Akhazi
Booked Day: Friday: 2023-06-16
```

(10 marks)