**NANYANG TECHNOLOGICAL UNIVERSITY**

**SEMESTER 2 EXAMINATION 2022-2023**

**SC2002/CZ2002– OBJECT ORIENTED DESIGN & PROGRAMMINGE**

Apr/May 2023 Time Allowed: 2 hours

**INSTRUCTIONS**

1. This paper contains 4 questions and comprises 8 pages.
2. Answer **ALL** questions.
3. This is a closed-book examination.
4. All questions carry equal marks.

---

1. (a) Give the output of the following code with brief justification.

```
public class SchoolBag {
    private String color;
    private double weight = 0;
    private double max_load;

    SchoolBag(String bagcolor, double bagmax_load){
        color = bagcolor;
        max_load = bagmax_load;
    }

    void load(double books){
        if ((weight + books) > max_load){
            System.out.print("Overload. cannot put in. ");
        }
        else{
            System.out.print("Put in successfully. ");
            weight = weight + books;
        }
    }
}
```

Note: Question No. 1 continues on Page 2

```
public class TestBag {

    public static void main(String[] args) {
        SchoolBag JustinBag = new SchoolBag(“black”, 15);

        JustinBag.load(11);
        JustinBag.load(6);
    }
}
```

(4 marks)

(b) Give the output of the following code with brief justification.

```
public class A {
     public static int s;
     public int ns;
}

public class TestA {
    public static void main(String[] args) {
        A a1 = new A();
        a1.s = 100;
        a1.ns = 99;
        A a2 = new A();
        System.out.println("a2 s = "+a2.s+" ns = "+a2.ns);
    }
}
```

(4 marks)

(c) Give the output of the following code with brief justification.

```
public class Animal {
    public void print(Animal a){
        System.out.print("Animal");
    }
}
class Cat extends Animal{
    public void print(Cat c){
       System.out.print("Cat");
    }
}
```

Note: Question No. 1 continues on Page 3

```
public class Test {
  public static void main(String[] args) {
        Animal a1 = new Cat();
        Cat a2 = new Cat();
        a2.print(a1);
  }
}
```

(5 marks)

(d) Explain the THREE ways of method overriding.

(5 marks)

(e) Explain the difference between object upcasting and object downcasting with examples. Discuss issues with the usage of them, if any.

(7 marks)

2. You are tasked to write a Java application program to create a simple video game that involves different types of vehicles such as car and boat. Different types of vehicles have different maximum speeds and their respective attributes. Players' scores are updated based on how well they perform in driving the vehicle.

(a) Write code for the `Score` class that keeps track of the players' scores in the game. It has the following attribute and methods:

- `score`: an integer representing the player's current score.
- `getScore`: returns the current score.
- `updateScore`: updates the score by adding 1 point.
- `resetScore`: resets the score to zero.
- `printScoreboard`: prints the current score to the console.

(5 marks)

(b) Write code for the abstract class `Vehicle` that has the following attributes and method:

- `name`: a string representing the name of the vehicle.
- `maxSpeed`: an integer representing the maximum speed of the vehicle.
- `drive`: an abstract method.

(5 marks)

Note: Question No. 2 continues on Page 4

(c) Two subclasses, `Car` and `Boat` are derived from the `Vehicle` class:

The `Car` class has two additional attributes:
- `numWheels`: an integer representing the number of wheels on the car.
- `numDoors`: an integer representing the number of doors on the car.

The `Car` class should override the `drive` method to print a message to the console that says 'Driving the car at max speed: [maxSpeed]'.

The `Boat` class has one additional attribute:
- `propulsion`: a string representing the type of propulsion used by the boat (e.g., 'motor', 'sail', and 'paddle')

The `Boat` class should override the drive method to print a message to the console that says 'Driving the boat with [propulsion] at max speed: [maxSpeed]'.
(Hint: [maxSpeed] and [propulsion] mean the placeholders of the values of the corresponding variable `maxSpeed` and `propulsion`, respectively.)

(10 marks)

(d) Write a class `Game` that has the following method:
- `start`: starts the game by calling the `drive` method with the given type of `Vehicle` in the argument of the method.

(5 marks)

3. Study the following description of a Camp Management System (CMS):

CMS is a web-based system that maintains a centralized repository of all related information. CMS allows clients to easily access the relevant information and book their camp arrangements.

To access CMS, clients must register with the system by providing their password, name, email address, residence address. After registration, clients can access CMS to book their camps by selecting the camp and paying the camp fee.

For each camp, information such as camp title, camp fee, start and end dates, instructor/facilitator details, and venue details is to be provided.

The venue details provide the information on the venue name, street, and postcode.

Note: Question No. 3 continues on Page 5

The instructor/facilitator details provide the instructor/facilitator's information such as name, gender, age, education background, working experience and photograph.

Only an admin, who is one of the instructors, can modify camp information.

Clients can enter promotion code to enjoy discount in camp fee. There are three promotion codes, LOYALTY, FIRSTTIMER, and REFER. The system will check client's eligibility of using a promotion code.

The discount rates for LOYALTY, FIRSTTIMER, and REFER are 5%, 2%, and 3%, respectively.

Only an admin, who is one of the instructors, can view the reports of camp booking through the web-based system. The report can be generated based on individual camp or individual user.

Clients can book multiple camps and they can view their own camp booking reports.

Currently, a client can pay the camp fee via Paypal, eNets, and credit cards. The system might provide more payment methods later.

(a) You are being tasked to identify the entity classes needed to build the CMS based on the description above. Show your design in a Class Diagram. Your Class Diagram should show clearly the relationship between classes, relevant attributes (at least TWO), multiplicities, association names, if any. You need not show the class methods unless they are clearly stated in the description.

(17 marks)

(b) In the application, the details of camp booking reports should allow to be printed in different formats like XML (eXtensible Markup Language), HTML (HyperText Markup Language), plain text, or any other format. For example, if the required format is to be in XML, both the person's and his/her achievements details will be generated in XML.

(i) In order to cater to different formats and minimize changes, you are advised to study and make use of the design principle(s) shown in Figure Q3a (on page 6). The design is useful in allowing an external class to provide additional functionality required by existing classes. Show in a Class Diagram how you can adapt the design in Figure Q3a to suit the requirement. Your Class Diagram should clearly show the class names, relevant attributes and methods to illustrate your ideas.

(4 marks)

Note: Question No. 3 continues on Page 6

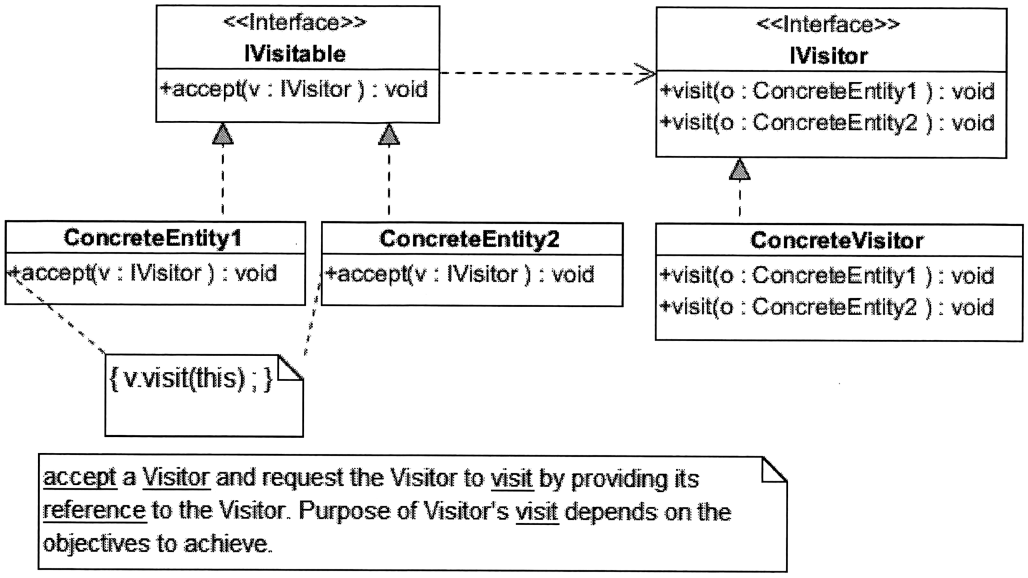


**Figure Q3a**

(ii) By using the design principles, give your comments of the design used in Q3b(i).

(4 marks)

4. You are required to write a **C++** application program to calculate and show a student's overall score for a course he/she took. A course has several components such as an assignment, a research report, a final exam, etc. Each course component takes a certain percentage when calculating the overall score of a student for the course. A student can be an undergraduate student or a postgraduate student. A course taken by an undergraduate student has only two components, including an assignment and a final exam. A course taken by a postgraduate student has an additional component of a research report.

(a) Write **C++** code for a `CourseComponent` class that has the private instance variables `componentName`, `percentage` and `score`, a constructor, and methods of `getPercentage` and `getScore`.

- `componentName`: the name of the course component.
- `percentage`: the percentage that the course component takes when calculating the overall score obtained by a student for the course.
- `score`: the score of the course component that a student obtains.
- the constructor initializes the values of all instance variables.
- `getPercentage`: accessor method to get the percentage value.
- `getScore`: accessor method to get the score value.

(5 marks)

Note: Question No. 4 continues on Page 7

(b) Write **C++** code for an abstract class `Student` that has an instance variable `name`, a constructor, and a method `calOverall`.

- `name`: the name of the student.
- the constructor initializes the name of the student.
- `calOverall`: an abstract method.

(5 marks)

(c) Two subclasses, `Undergraduate` and `Postgraduate`, are derived from the `Student` class. `Undergraduate` class has two course components of `assignment` and `finalExam`. `Postgraduate` class has three course components of `assignment`, `researchReport` and `finalExam`. Both classes implement the method of `calOverall` that calculates the overall score of a course obtained by a student, which is the weighted score of all course components. The component percentages are demonstrated in Table Q4 as follows:

**Table Q4**

| | `Undergraduate` | `Postgraduate` |
|---|---|---|
| `assignment` | 40% | 20% |
| `researchReport` | N/A | 20% |
| `finalExam` | 60% | 60% |

Write **C++** code for the two subclasses.

(8 marks)

(d) Write studentMain.cpp to have the `printOverall` method which prints out the overall score of a course taken by a student by calling the method `calOverall`. Demonstrate **dynamic binding** of the `calOverall` in `printOverall` method for an undergraduate student and a postgraduate student. (Hint: having a single `printOverall` method in studentMain.cpp, regardless of which `Student` subclass passed as argument of the `printOverall` method).

(7 marks)

**Note**: Separate the declaration code and implementation code – create a header file/s (.h) for the declaration code and implementation file/s (.cpp) for the implementation code.

Note: Question No. 4 continues on Page 8

A sample run of the student system is shown below:

Please choose student category (U- Undergraduate, P-Postgraduate, E-Exit): **P**
Student name: **Alice**
Assignment score: **80**
Report score: **70**
Final exam: **90**
Alice--overall score is 84.

Please choose student category (U- Undergraduate, P-Postgraduate, E-Exit): **U**
Student name: **Tom**
Assignment score: **80**
Final exam: **90**
Tom--overall score is 86.

Please choose student category (U- Undergraduate, P-Postgraduate, E-Exit): **E**
Thank you for using the system. Bye.



END OF PAPER

**CZ2002 OBJECT ORIENTED DESIGN & PROGRAMMING**
**SC2002 OBJECT ORIENTED DESIGN & PROGRAMMING**

Please read the following instructions carefully:

1. **Please do not turn over the question paper until you are told to do so. Disciplinary action may be taken against you if you do so.**

2. You are not allowed to leave the examination hall unless accompanied by an invigilator. You may raise your hand if you need to communicate with the invigilator.

3. Please write your Matriculation Number on the front of the answer book.

4. Please indicate clearly in the answer book (at the appropriate place) if you are continuing the answer to a question elsewhere in the book.