TTTK1143 - Program Design and Problem Solving
Tutorial 1

Part A

Topic: Object

1. Identify THREE objects in the real world that must be included in Student Information System, such as smpweb.

2. State THREE characteristics of an object.

3. State whether each of the objects below is TANGIBLE or INTANGIBLE.
   a) Programmer
   b) Computer
   c) Programme
   d) Software

Topic: Object State and Attributes

4. Name at least THREE attributes that are appropriate for each of the following objects:
   a) Lecturer
   b) Department

5. Give a value to the TWO attributes identified in (4).

6. Show an example of TWO objects in (4) with different states – meaning the value of their attributes are different.

Topic: Object Behaviour and Methods

7. Choose ONE object in (4), and identify TWO methods of the object.

Topic: Object Identity

8. State whether the following statement is TRUE or FALSE. Justify your answer.

   "Two objects of the same type/class will have the same identity."

Topic: Object Reference Variable

9. Write a Java statement to declare an object reference variable named myPet to reference a Animal object.

Topic: Object Creation/Instantiaton

10. Write a Java statement to create a Animal object, and assign the object to the object reference variable in (9).

11. How many Computer objects exist after the execution of the following Java statements?

```
Computer compOne = new Computer();
Computer compTwo = compOne;
Computer compThree = new Computer();
```

12. How many Car objects exist after the execution of the following Java statements?

```
Car carOne = new Car();
Car carTwo = carOne;
Car carThree = carTwo;

Car carFour = new Car();
Car carFive = carOne;
```

Topic: Class

13. Define the relationship between object and class in Object-Oriented Programming (OOP).

14. Name at least three attributes that are appropriate for each of the following classes:
    a. Classroom
    b. CreditCardBill

Topic: Constructor

15. What is the difference between constructor method and method?

16. Define a default constructor that will be inserted by the Java compiler into a class named Student if no constructor is defined for the class by the programmer.

17. Consider the following class definition.

```
public class Student {

  private String matricNo;
  private String name;
  private int age;

  public Student(String m, int  a) {
     matricNo = m;
     name = "Unspecified";
     age = a;
  }

public Student(String m, String n, int a) {
  matricNo = m;
  name = n;
  age = a;
}

public void displayInfo() {
  System.out.println("MatricNo: " + matricNo);
  System.out.println("Name: " + name);
  System.out.println("Age: " + age);
}
}
```

(a) Write the output of the following statements.

(i)
```
Student student1 = new Student ("A123455", "Waiz", 22);
student1.displayInfo();
```

(ii)
```
Student student1 = new Student("A123457", 21);
student1.displayInfo();
```

(b) State the reason why the following statement will result in a compilation error.

```
Student myStudent = new Student("A123558", "Wafiy");
```

Topic: Object Members – Instance Variables

18. Choose the MOST APPROPRIATE access control (or visibility modifier) that should be assigned to an instance variable of a class.
    a) public
    b) private
    c) default

19. Consider the following class definition.

```
public class Student {
  private String matricNo;
  String name;
  public double age;
}
```

State whether the following statements are ALLOWED or NOT ALLOWED. Justify your answers.

```
public class TesterListStudent {

    public static void main(String[] args) {
      Student student1 = new Student();
      System.out.println(student1.matricNo); // Line 1
      System.out.println(student1.name); // Line 2
      System.out.println(student1.age); // Line 3
    }
}
```

(a) Line 1
(b) Line 2
(c) Line 3

Topic: Object Members – Accessor and Mutator Methods

20. Is each of the following method identifiers (a) legal and conventional, (b) legal but unconventional, or (c) illegal?
    a. void()
    b. 36542ZipCode()
    c. associationRules()
    d. displayTotal()
    e. Accounts_Receivable()
    f. Invoice()

21. Consider the following class definition.

```
public class Invoice {
  private int idInvoice ;
}
```

a. Define an accessor method for the Invoice class, to allow the value of its instance variable brand to be read from outside of the class.

b. Define a mutator method for the Invoice class, to allow the value of its instance variable brand to be modified from outside of the class.

Topic: Class Members – Static (or class) Variables/Methods vs. Non-static (or instance) Variables/Methods

22. Consider the following class definition.

```
public class Course {
  public String id;
  public  String name;
  public static int noOfStudent = 5;

  public Course(String id, String name) {
     this.id = id;
     this.name = name;
     noOfStudent++;
  }

  public String getId() {
     return id;
  }

  public String getName() {
     return name;
  }

  public static int getNoOfStudent() {
     return noOfStudent; }
  }
```

(a) Write the output of the following statements.

```
public class TestCourse {
  public static void main(String[] args) {

     Course course1 = new Course("TK1114", "Java Programming I");
     System.out.println("ID: " + course1.id);
     System.out.println("Course Name : " + course1.name);
     System.out.println("Number of students: " +
                                          course1.noOfStudent);

     Course course2 = new Course("TK1143", "Java Programming II");
     System.out.println("ID: " + course2.id);
     System.out.println("Course Name : " + course2.name);
     System.out.println("Number of students: " +
                                          course2.noOfStudent);
  }
}
```

(b) State whether the following statements are ALLOWED or NOT ALLOWED. Justify your answers.

```
public class TestCourse {

 public static void main(String[] args) {
    System.out.println(Course.noOfStudent); // Line 1
    System.out.println(Course.id); // Line 2
    System.out.println(Course.name); // Line 3
    System.out.println(Course.getNoOfStudent()); // Line4
    System.out.println(Course.getId()); // Line 5
 }
}
```

(i) Line 1
(ii) Line 2
(iii) Line 3
(iv) Line 4
(v) Line 5

Part B – Problem Solving

Problem Description

Si Penyu Sdn Bhd starts its own online shop since 2015 to fulfill customer needs and technology demand. The company strategies is to provide the best deals to gain customer's support. Customers who register for its Reward Plan with free Happy Card will get 1 point for every RM 1 spent at the online shop.

The customers also get special offer for the following purchase:

- RM100-RM499 : 5 times the points collected
- >=RM500 : 7 times the point collected

The points collected by registered customer can be redeemed to pay for items bought from the online shop. Each member can have only one Happy Card, and the type of the card is categorised into four categories based on the current total of the collected points. For newly registered customers who have not made any purchase the card type status is "New".

- Below 500 : Bronze
- 500 – 2499 : Silver
- 2500- 5499 : Gold
- Above 5500 : Platinum

Your Task

Write definition for class HappyCard and Customer in Java based on the following class description.

- Description for class HappyCard .

Class: HappyCard	Description
- point: int - type: String	Represents card's points. Represents card's type.
+ HappyCard() + addPoint(int): void + deductPoint(int): boolean - updateType(int): void + getPoint(): int + getType(): String	Constructor to initialise card's points as 0 and type as "New". Adds card's points and update card's type using `updateType` method. Deducts card's points and update card's type using `updateType` method, and returns true if amount to be deducted is valid. Returns false if otherwise. Updates card's type based on current points. Returns card's points. Returns card's type.

- Description for class Customer.

Class: Customer	Description
- customerID: int	Represents customer's ID.
- name: String	Represents customer's name.
- card: HappyCard	Represents customer's Happy card.
+ Customer(String, String)	Constructor to assign customer's ID and name, as well as create customer's `HappyCard` object.
+ buy(int): void	Asks customer's card to add points.
+ redeem(int): boolean	Asks customer's card to deduct points.
+ getId(): String	Returns customer's id.
+ getName(): String	Returns customer's name.
+ getCard():HappyCard	Returns customer's `HappyCard` object.

- Code for class TestRewardPlan that uses class Customer and HappyCard.
- Please fill in the blanks to complete the tester code

```
import java.util.*;

public class TestRewardPlan{
   public static void main (String[] arg){
      Scanner sc = new Scanner (System.in);

       System.out.println("++++++ Customer Redeem Plan +++++\n");
       System.out.print(" Customer ID   >> ");
       int id = sc.nextInt();
       sc.nextLine();
       System.out.print(" Name          >> ");
       String name = sc.nextLine();

       //creating objects ???

       System.out.print("\n**OPERATIONS : \n1. Buy Product  2. Redeem  3.Display
                        Details 0. Exit >> ");
       int ops = sc.nextInt();

       while (ops != 0) {
          if (ops == 1) {
             System.out.print("\nTotal Ringgit to buy (whole number only)    >>  ");
             int amount = sc.nextInt();

             customer.buy(amount);

             System.out.println("\n** Your points have been added.");
             System.out.println("Current points: " + card.getPoint());
          }

          else if (ops == 2) {
             System.out.print("\nTotal Ringgit to redeem (whole number only)  >>  ");
             int amount = sc.nextInt();

             if (customer.redeem(amount)) {
                System.out.println("\n** You have successfully redeemed your
                                    points.");
```

```
                System.out.println("Current points: " + card.getPoint());
            }

            else {
                System.out.println("\n## Transaction failed ##");
                System.out.println("Sorry, your points are not enough for
                                    redemption.");
            }
        }

        else if (ops == 3) {
            System.out.println("\nCustomer ID: " + customer.getId());
            System.out.println("Name: " + customer.getName());
            System.out.println("Current points: " + card.getPoint());
            System.out.println("Card type: " + card.getType());
        }

        else if (ops > 3)
            System.out.println("Invalid input");

        System.out.print("\n**OPERATIONS : \n1. Buy Product  2. Redeem  3.Display
                          Details 0. Exit >> ");
            ops = sc.nextInt();
        }

        sc.close();
    }
}
```

- Sample output

Case 1 : Customer choose display points operation

```
++++++ Customer Redeem Plan +++++

 Customer ID   >> 2451
 Name          >> Kelly

**OPERATIONS :
1. Buy Product  2. Redeem  3.Display Details 0. Exit >> 3

Customer ID   : 2451
Name          : Kelly
Current points: 0
Card type     : New
```

Case 2: Member choose buy product operation for enter amount and get the points

```
**OPERATIONS :
1. Buy Product  2. Redeem  3.Display Details 0. Exit >> 1

Total Ringgit to buy (whole number only)      >>  55

** Your points have been added.
Current points: 55

**OPERATIONS :
1. Buy Product  2. Redeem  3.Display Details 0. Exit >> 1
```

```
Total Ringgit to buy (whole number only)      >>  100

** Your points have been added.
Current points: 555

**OPERATIONS :
1. Buy Product  2. Redeem  3.Display Details 0. Exit >> 3

Customer ID   : 2451
Name          : Kelly
Current points: 555
Card type     : Silver
```

Case 3: Member choose Redeem operation and unsuccessfully redeem

```
++++++ Customer Redeem Plan +++++

 Customer ID   >> 1234
 Name          >> John

**OPERATIONS :
1. Buy Product  2. Redeem  3.Display Details 0. Exit >> 3

Customer ID   : 1234
Name          : John
Current points: 0
Card type     : New

**OPERATIONS :
1. Buy Product  2. Redeem  3.Display Details 0. Exit >> 2

Total Ringgit to redeem (whole number only)  >>  30

## Transaction failed ##
Sorry, your points are not enough for redemption.
```

Case 4: Member choose Redeem operation and successfully redeem

```
++++++ Customer Redeem Plan +++++

 Customer ID   >> 1234
 Name          >> John

**OPERATIONS :
1. Buy Product  2. Redeem  3.Display Details 0. Exit >> 2

Total Ringgit to redeem (whole number only)  >>  70

** You have successfully redeemed your points.
Current points: 430
```

Case 5: Member choose buy products and get points times five and upgrade category

```
++++++ Customer Redeem Plan +++++

 Customer ID   >> 4442
 Name          >> Travolta

**OPERATIONS :
1. Buy Product  2. Redeem  3.Display Details 0. Exit >> 3

Customer ID   : 4442
Name          : Travolta
Current points: 0
Card type     : New

**OPERATIONS :
1. Buy Product  2. Redeem  3.Display Details 0. Exit >> 1

Total Ringgit to buy (whole number only)      >>  700

** Your points have been added.
Current points: 4900

**OPERATIONS :
1. Buy Product  2. Redeem  3.Display Details 0. Exit >> 3

Customer ID   : 4442
Name          : Travolta
Current points: 4900
Card type     : Gold

**OPERATIONS :
1. Buy Product  2. Redeem  3.Display Details 0. Exit >> 1

Total Ringgit to buy (whole number only)      >>  1000

** Your points have been added.
Current points: 11900

**OPERATIONS :
1. Buy Product  2. Redeem  3.Display Details 0. Exit >> 3

Customer ID   : 4442
Name          : Travolta
Current points: 11900
Card type     : Platinum
```

Case 6: Member choose other of listed menu

```
++++++ Customer Redeem Plan +++++

 Customer ID   >> 1234
 Name          >> John

OPERATIONS :
1. Buy Product  2. Redeem  3. Display Points >> 4

Invalid Input
```