Object Oriented Programming – SCJ2153

Static Class Members

Associate Prof. Dr. Norazah Yusof

Static Class Member

- A static class member belongs to the class, not to the objects instantiated from the class.
- Things that you can mark static:
  - Static variables
  - Static methods
- The `static` modifier is used to create static variables and methods that will exist independently of any instances created for the class.

Reasons for using Static Member

1. Need to count all instances instantiated from a particular class.
2. To share data among all the instances of a class.
3. To have a method that always run the same way without having dependency on the instance variable values.

Static Modifier

- Things you can't mark as static:
  - Constructors
  - Classes (unless they are nested)
  - Interfaces
  - Inner class methods and instance variables
  - Local variables

Static Variable

- All object of the same class are affected if one object changes the value of a static variable.
  - Changing the value of one static variable in one object changes it for all others.
- can be used to collect statistics or totals for all objects of the class (for example, total sales for all vending machines)
- Declaration example of a private static variable:

```
private static int count = 0;
```

- A public static variables are referred to using “dot notation” (if calling from another classes):

```
ClassName.staticVar
```

Static constant

- A constant can also be declared as a static constant – the constant can be shared by all members of the class:

```
public class Product
{
   static final double MAX_RATE = 0.9;
   static final double MIN_RATE = 0.04;
   ...

}
```

Static Methods

- Static methods can access and manipulate a class’s static attributes.
- Static methods **cannot** access non-static attributes or **cannot** call non-static methods of the class.
- From outside the class, static methods are called using “dot notation”:
  `ClassName.staticMethod()`
- Exam

```
double x = Math.random();
double y = Math.sqrt (x);
System.exit();
```

Programming Example

- Extend the Travel program so that it can keep track of the total travellers being created and the total payment that had been made by all the travellers.

  Hint: use static variables and static methods

Travel.java

```java
public class Travel {
  private String destination,name,address;
  private double price;
  private static int countTraveller=0; //track traveller
  private static double totalPay=0;  //track total payment

  public Travel() {
   countTraveller++; //increment traveller
  }

  public Travel(String name,String destination){
    this.name = name ;
    this.destination = destination;
    countTraveller++; //increment traveller
  }

  public String getDestination(){
    return destination;
  }
```

Travel.java (cont.)

```
public String getName(){
    return name;
  }

  public String getAddress(){
    return address;
  }

  public double getPrice(){
    return price;
  }

  public static int getCountTraveller() {
    return countTraveller;
  }

  public static double getTotalPay() {
    return totalPay;
  }
```

Travel.java (cont.)

```
  public void setDestination(String d){
    destination = d;
  }

  public void setName(String n){
    name = n;
  }

  public void setAddress(String a){
    address = a;
  }

  public void setPrice(double p){
    price = p;
    totalPay += price; //accumulate total payment
  }
```

Travel.java (cont.)

```
  public void display(){
    System.out.println("\nDestination = "+destination);
    System.out.println("Name of traveller = "+name);
    System.out.println("Address of traveller = “ +
      address);
    System.out.printf("Price = RM %.2f \n", price);
  }
}
```

TestTravel.java

```
public class TestTravel {
  public static void main(String[] args){
     Travel ob1 = new Travel();
     ob1.setName("Linda");
     ob1.setDestination("Indonesia");
     ob1.setAddress("NO.1 Tmn Impiana,81100 Johor Bahru,
Malaysia");
     ob1.setPrice(999);
     ob1.display();

     Travel ob2 = new Travel("Fazlin","Hawaii");
     ob2.setAddress("No. 2 Tmn Bahagia, 84000 Muar,
Malaysia");
     ob2.setPrice(4999);
     ob2.display();
```

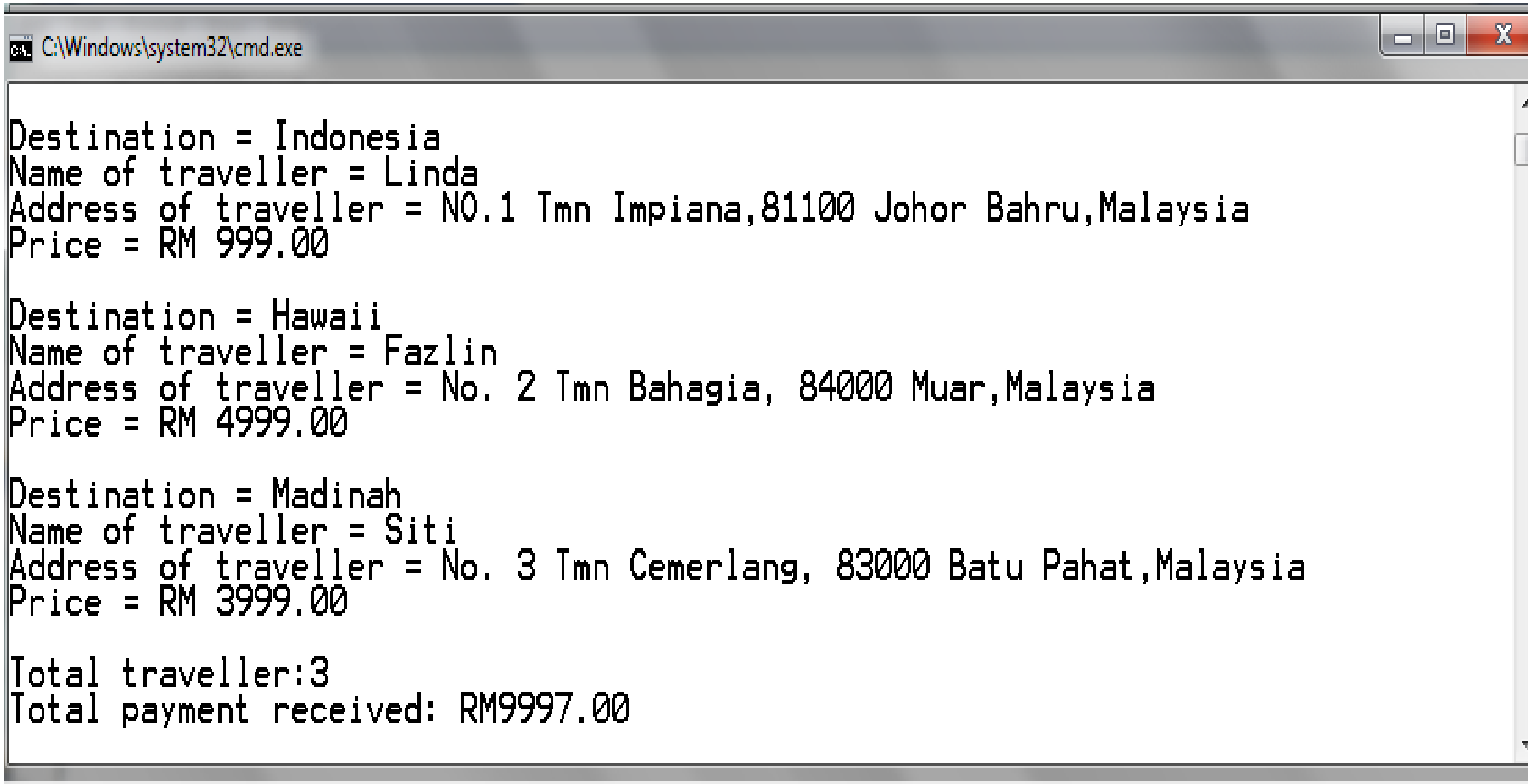
TestTravel.java (cont.)

```
     Travel ob3 = new Travel("Siti","Madinah");
     ob3.setAddress("No. 3 Tmn Cemerlang, 83000 Batu
       Pahat, Malaysia");
     ob3.setPrice(3999);
     ob3.display();

     System.out.println();
     System.out.println ("Total traveller:" +
      Travel.getCountTraveller()); //invoke static method
     System.out.printf ("Total payment received: RM%.2f",
      Travel.getTotalPay()); //invoke static method
     System.out.println();
   }
}
```

TestTravel.java (cont.)

Output:



```
Destination = Indonesia
Name of traveller = Linda
Address of traveller = NO.1 Tmn Impiana,81100 Johor Bahru,Malaysia
Price = RM 999.00

Destination = Hawaii
Name of traveller = Fazlin
Address of traveller = No. 2 Tmn Bahagia, 84000 Muar,Malaysia
Price = RM 4999.00

Destination = Madinah
Name of traveller = Siti
Address of traveller = No. 3 Tmn Cemerlang, 83000 Batu Pahat,Malaysia
Price = RM 3999.00

Total traveller:3
Total payment received: RM9997.00
```