The University of the West Indies, St. Augustine
COMP 2603 Object Oriented Programming 1
Lab 10

Learning Objectives

- Create and use a HashMap to store, retrieve, remove and update objects
- Create and use a TreeMap to store, retrieve, remove and update objects

Create a new BlueJ Project called Lab10. Add the Java files provided on myElearning to your project.

Part 1: Create Passenger objects

(a) Create a main class, Lab10, containing 5 Passenger objects with the following names:

```
Passenger p1 = new Passenger("Joe");
Passenger p2 = new Passenger("Sid");
Passenger p3 = new Passenger("Lou");
Passenger p4 = new Passenger(“Gil");
Passenger p5 = new Passenger("Moe");
```

(b) Add the passengers to a **TreeMap** called **passengers** using the passenger name as the key and the passenger object as the value.

(c) Print out the <key,value> pairs in the passengers map

```
System.out.println(passengers);
```

(d) Write code to print Gil’s the ticket number.

Answer:

(e) Add the following code to update Sid’s name to Syd and retrieve Syd’s ticket number.

```
p2.setName("Syd");
Passenger syd = passengers.get("Syd");
Ticket sydsTicket = syd.getTicket();
System.out.println(sydsTicket);
```

(i) Why is the error thrown?

Answer:

(ii) What must be done in order to make the code work? Modify accordingly.

Answer:

Part 2: Create Vehicle objects

In the Lab12 main class:

(a) Create 5 **Vehicle** objects using the following code

```
for(int i = 0; i<5; i++){
   Vehicle v = new Vehicle(getRandomNumber(1,20),
                                   getRandomNumber(1,5),
                                   getRandomNumber(1,5),
                                   getRandomNumber(1,5)));
   System.out.println(v);
}
```

(b) Edit the code from 2(a) so that 5 **Vehicle** objects are created with the following plateIDs: RLM01, CTJ02, DSC03, MYA04, BTN05. These correspond to objects v1, v2, v3, v4, and v5.

(c) Create a **TreeMap** called **vehicles** that stores <Vehicle, Passenger> key-value pairs. Add the following mapping and print out the map.

Vehicle	Passenger
RLM01	P1

(i) Did it work? Why not?

Answer:

(ii) What is needed to make the code work?

Answer:

(d) Add the following mappings and print out the map.

Vehicle	Passenger
MYA04	P3
CTJ02	P5

What do you notice about the order of the mappings? How is this order achieved?

Answer:

(e) Add the following code after part (d) above.

```
Vehicle v6 = new Vehicle
("CTJ02",getRandomNumber(1,5),getRandomNumber(1,5),getRandomNumber(1,5));
        manifest.put(v6, p2);
        System.out.println("Part 2(e)\n" +manifest);
```

(i) What do you notice about the contents of the map? Why did this happen?

Answer:

(ii) Is it possible to have Vehicle object v6 and v2 used as keys in the map?Why or why not?

Answer:

Part 3: Store Ticket and Passenger objects in a HashMap

In the Lab12 main class:

(a) Create a **HashMap** called **ticketList** that stores <Ticket, Passenger> key-value pairs

(b) Add the five passengers from 1(a) to the HashMap and printout the ticketList contents.

(c) Create a new Ticket object tx with the ID 100.
  (i) Try to insert it into the ticketList for passenger p3 (Lou). Did this work? Why?

```
Ticket tx = new Ticket();
tx.setTicketNumber(100);
ticketList.put(tx, p3);
System.out.println("Part 3(c)(i)\n" +ticketList);
```

Answer:

  (ii) Try to remove Ticket object with ID 100 as a key from the ticketList using the code below.Did this work? Why or why not?
  (iii)

```
Ticket ty = new Ticket();
 ty.setTicketNumber(100);
 ticketList.remove(ty );
    System.out.println("Part 3(c)(ii)\n" +ticketList);
```

Answer:

(iii)Try to determine whether the ticketList contains a key object with ID = 100 using the code below:

```
Ticket tz = new Ticket();
tz.setTicketNumber(100);
System.out.println("Part 3(c)(iii)\n”+ ticketList.containsKey(tx ));
System.out.println("Part 3(c)(iii)\n"+ ticketList.containsKey(ty ));
System.out.println("Part 3(c)(iii)\n"+ ticketList.containsKey(tz ));
```

Did this work? Why not?

Answer:

(d) Add a hashCode( ) and equals( ) method to the Ticket class where both work using the Ticket ID. How does the behaviour in parts C i, ii , iii differ now?

Answer: