

Question 1 (7 marks)

```
1  Car fleet[] = new Car[4];
2  fleet[0] = new Car();
3  fleet[1] = new Car();
4  fleet[2] = new Car();
```

- Assuming
 - The Car class has been correctly coded with 2 instance variables both with an accessor and mutator pair of methods
- After the code has executed:
 - a) How many Car instances?
 - b) How many objects? Briefly explain
 - c) What is the value of fleet[0]?
 - d) Where is the value of fleet[0], stack or heap?
 - e) Is there a problem with fleet[3]? If so what?
 - f) Is there a problem with fleet[4]? If so what?
 - g) Is there a problem with fleet[1].setMaker("BMW");? If so what?

a. 3

The statement "new Car ();" is used to create an instance of the Car class. Because in line 2.3.4, there are 3 "new Car ();" in total, 3 instances of the Car class have been created. Therefore, there are 3 Car instances.

b. 4

In line 1, "Car fleet [] = new Car [4];" declares an array called fleet. As arrays are objects, we have an object fleet after the declaration. The statement "new Car ();" is used to create an instance of the Car class. Because in line 2.3.4, there are 3 "new Car ();" in total, 3 instances of the Car class have been created. As an instance of a class is an object, we have 3 Car instances, which means that we have 3 Car objects. Therefore, there are 4 objects in total.

c. "fleet[0]" stores the address of a Car object, so the value of fleet[0] is the address of the Car object.

d. heap

An object's data are kept in heap. As arrays are objects, the value of the array element "fleet[0]" is stored in heap.

e. No

In line 1 when declaring the array, the length is 4 (which specifies the fixed number of element in the array). In Java, the first element of an array has the index 0 and the last element of an array has the index of length-1. Therefore, in the fleet array, the elements have index 0, 1, 2, 3. Therefore, there will not be any problem with fleet[3].

f. Yes

In line 1 when declaring the array, the length is 4 (which specifies the fixed number of element in the array). In Java, the first element of an array has the index 0 and the last element of an array has the index of length-1. In addition, Java arrays cannot be resized or oversized. Therefore, in the fleet array, the elements only have index 0, 1, 2, 3. In other words, the fleet array cannot have the index 4. Therefore, fleet[4] will result in an error.

g. Generally, yes, there is a problem.

If there is not a method called setMaker in the Car class, an error occurs when executing the statement "fleet[1].setMaker ("BMW");". This is because methods have to be defined before they are used. If we did not define the method setMaker in the Car class, we cannot use it. However, if we have defined a method called setMaker in the Car class, then we can use this method to perform an operation on the fleet array element (whose element type is Car). Therefore, there are two different situations. Because we do not know what instance variables with accessor and mutator pair have been coded in the Car class, we assume there can be a problem with "fleet[1].setMaker ("BMW");".

No, there is no problem.

Question 2 (2*6=12 marks)

Assume you have the following integer values:4,90,3,11. Write a piece of code that executes the following requirements in sequence:

- a) Creates an array list of integers
 - b) Add the four values to the list
 - c) Print the list
 - d) Insert value (5) in the second cell
 - e) Delete the last cell
 - f) Delete all the elements
-

```
import java.util.ArrayList;
public class Test {
    public static void main(String[] args) {
        //create an array list
        ArrayList arrList = new ArrayList<>();
        //add the values 4,90,3,11 to the list
        arrList.add(4);
        arrList.add(90);
        arrList.add(3);
        arrList.add(11);
        //print the list
        System.out.println(arrList);
        //insert value 5 in the 2nd cell
        arrList.add(1,5);
        //delete the last cell
        arrList.remove (arrList.size()-1);
        //delete all elements
        arrList.clear();
    }
}
```

After executing this code, the output is:
[4, 90, 3, 11]

```
import java.util.ArrayList;
public class Test {
    public static void main(String[] args) {
        //create an array list
        ArrayList<Integer> arrList = new ArrayList<>();
        //add the values 4,90,3,11 to the list
        arrList.add(4);
        arrList.add(90);
        arrList.add(3);
        arrList.add(11);
        //print the list
        System.out.println(arrList);
        //insert value 5 in the 2nd cell
        arrList.add(1, 5);
        //delete the last cell
        arrList.remove(arrList.size() - 1);
        //delete all elements
        arrList.clear();
    }
}
```