Software and Programming 1

Lab 6:
Use of classes,
arrays and debugger

11 February 2015
Exercise 1

1. What output is produced when the following Java program is executed? The code is available from http://www.dcs.bbk.ac.uk/~roman/sp1额外/T09.java

Execute the program step-by-step in the debugger.
Class T09

• Launch BlueJ.

• Create a new Project and give it a suitable name, e.g. SP1Lab6.

• Create a new class and enter the name T09.

• Copy the code which is available from http://www.dcs.bbk.ac.uk/~roman/sp1/extra/T09.java into the new class, T09.
public class T09
{
    private int i = 1;
    private static int v = 2;

    public T09()
    {
        System.out.println("constructor of T09: i = "+i+",
                          v = " + v);
        i = 0;
        v++;
    }
}
Step-by-step execution of the class T09 using a debugger (2)

```java
public static void main(String[] args) {
    T09 t1 = new T09();
    System.out.println("t1: i = " + t1.i + ", v = " + t1.v);
    t1.v = 7;
    T09 t2 = new T09();
    System.out.println("t2: i = " + t2.i + ", v = " + t2.v);
    t1.v = 5;
    System.out.println("t2: i = " + t2.i + ", v = " + t2.v);
}
```

What output is produced when the above program is executed?
Exercise 2

2. Implement a class **Cycle** for modelling cycle hire. Each instance of the class should store cycle's ID and a number of the docking station it is parked at (0 if it is being used).

The class should include:

- a constructor that automatically assigns an ID to a newly created instance of Cycle.
- instance methods - `getID`, `pickup` and `park`.
- a class method `getNumberOfCycles`. 
Instance methods:

- **getID** should return cycle's ID.

- **pickup**, with no parameters, should check whether the cycle is parked and, if it is available for hire, mark it as being used and return true; otherwise, it should return false.

- **park**, with a docking station ID as its parameter, should check whether the cycle is currently hired and if so, mark it as docked at the provided station ID and return true; otherwise, it should return false.

- **getNumberOfCycles** should return the number of cycles in the system.
Private data:

• Two instance variables
  - ID which represents a unique ID of the cycle.
  - dockingStationID which represents an ID of the docking station if the cycle is parked or 0 if it is in use.

• One class variable
  - lastAssignedNumber which stores the last number assigned to a newly created instance of Cycle.
public class Cycle
{
    /* To Do: declare private data, i.e. instance variables (ID and dockingStationID) and class variable (lastAssignedNumber) */

    /* Instance methods (public interface) */
    public int getID() { /* To Do */ }
    public boolean pickup() { /* To Do */ }
    public boolean park(int dockSID) { /* To Do */ }
    public int getNumberofCycles() { /* To Do */ }
/* Constructor */

public Cycle()
{

    /* To Do: */
    /* To Do: */
    (i) Assigns an ID to a newly created instance of Cycle
    (ii) Initialise the dockingStationID

}

} // end of class Cycle
Exercise 2 (3)

Implement another class **CycleTest** that declares and creates an array of 10 Cycles.

It then fills up the array by creating 10 instances of cycles and parks them at the docking stations specified by the user input.
import java.util.Scanner;

public class CycleTest
{

    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);

        /* To Do: 
        (i) Declare and create an array of 10 Cycles. 
        (ii) Fills up the array by creating 10 instances of cycles and parks them at the docking stations specified by the user input. 
        */
/* To Do:

Write additional code to fully test the remaining methods: getID, pickup and getNumberOfCycles

*/