Institute of Technology Tallaght, Department of Computing

B.Sc. (hons.) in Computing

2018/2019

C# Exercise

**C#**

You are required to write a program which assists in the administration of a set of servers in a data centre, some Windows, some Linux.

**Tasks:**

1. Design and develop an abstract class Server to represent a generic server machine:
   1. A Server object has a server name, add a field and corresponding read-write property for the name and enforce validation (i.e. name must be between 1 and 15 characters long, where each character is a – z or A – z or 1 - 9). Access to the set block should only be allowed within the class.
   2. A server has a non-null IP address (use System.Net.IPAddress), add a field and read/write property for this.
   3. Add a constructor which initialises the name and IP address to values supplied in parameters to the constructor; don’t have a default constructor in the class.
   4. Add an abstract method Restart with a String return value and takes no parameters – it will be used to restart a server and return log information about the restart.
   5. Implement the IComparable<T> interface to allow comparison of server objects for sort purposes by IP address i.e. 192.168.10.1 would be before 192.168.10.2 in sort order.
2. Design and develop 2 concrete subclasses of Server

LinuxServer

* 1. LinuxServer should have a “Distro” property (to indicate if it is Debian, Ubuntu, SUSE, RedHat, CentOS, or Fedora), use an auto read-only property.
  2. Add a non-default constructor which also sets the value of the Distro property
  3. Override ToString to return full information about the Linux server
  4. Override Restart to return a log like this "Restarting Linux [LinuxServer1] 192.168.10.2 Debian with / sbin / shutdown - r now"

and WindowsServer

1. Add a non-default constructor
2. Override ToString to return full information about the Windows server
3. Override Restart to return a log like this "Restarting Windows [WindowsServer1] 192.168.10.3 with rundll32.exe shell32.dll,SHExitWindowsEx 2 "
4. Add a ServerFarm class to represent a server farm
   1. Add a collection to store servers in the server farm (any type of server)
   2. Add a method which adds a new server into the farm once the IP address it uses is not already being used by a server in the farm
   3. Implement a restart method for the server farm which restarts each server in the farm in order from lowest to highest IP address (hint: sort the collection first). Aggregate the log from each server restart into one log and return it.
5. Implement 2 unit tests which test some aspect of the application

**Deliverables**:

1. ZIP project and upload to Moodle
2. Demo of code running or whatever has been achieved
3. Hand in hard copy of source code