

## questions-150605

**Advanced Programming (A.Y. 2014-2015)**

M.Sc. Engineering in Computer Science – Sapienza University of Rome

**June 5, 2015 – Duration 2h 00'**

- **Be concise and right to the point.** What you write should be understandable by a colleague of yours who just enrolled the M.Sc. in Engineering in CS.
- **Parts A and B MUST be written on separate sheets of paper.**
- **Each sheet should have your name written on it – Good luck!**

|                    |                     |                     |
|--------------------|---------------------|---------------------|
| <b>Last:</b> ..... | <b>First:</b> ..... | <b>Matr.:</b> ..... |
|--------------------|---------------------|---------------------|

**Part A****Question A1 (REST)**

Describe what is a RESTful Web service, by highlighting the peculiarities of this approach, also compared with more traditional SOAP Web service. Imagine you have to design a Web service offering information about cars and owners of such cars (e.g., the Web service offered by a Government authority that registers who owns cars). Provide the pseudo-interface of the SOAP Web service and of its RESTful counterpart, and comment them and your design choices.

**Question A2 (SCRUM)**

Describe the basic elements of SCRUM, by focusing on the concepts of sprint and backlog.

**Part B****Question B1 (MapReduce)**

1. Briefly describe the role of partitioners and combiners in the Hadoop MapReduce framework.
2. Design a MapReduce algorithm that computes the average of a set of  $n$  numbers. Provide the pseudo-code for the map and the reduce operations. You may use combiners/partitioners if you need them, but this is not mandatory. Make the size of the lists received by each reducer as small as you can, while keeping the number of rounds constant. You may assume that the algorithm knows  $n$  in advance.

**Question B2 (C++)**

1. Overload operator << such that the following program prints (5,7):

```
struct pair { int a, b; };
int main() {
    pair p = {5,7};
    std::cout << p << std::endl;
    return 0;
}
```

Notice that `std::cout` has type `std::ostream`.

2. Define a class `point` such that the following program fragment can be compiled:

```
point<double> p = { 5.2, 7.3 };
cout << p.x << " " << p.y << endl;
```

---

Publicato da [Google Drive](#) – [Segnala una violazione](#) – Aggiornato automaticamente ogni 5 minuti

---