COSC 1252-1254 Programming using C++
Assignment 1

Deliverable #1 **(5 marks)**

Due date: August 5, 2012; 23:59hrs
Late submission: August 10, 2012; 23:59hrs

1 Problem

Parasol Co., a pharmaceutical industry, has developed a new kind of virus capable of reviving dead cells. However, concerned with this virus causing a zombie outbreak, they decided to study the survival rate of people under such a situation. They have hired you to build a program that helps them to find out the probability of a person to survive such an outbreak, and which factors can be provided to people in such a disasterous situation. The simulation is divided in three deliverables, and this document briefs you the requirements of the first deliverable. Rather than being paid for the simulations, they spoke with your lecturer and he agreed to award you marks for each requirement successfully implemented.

As you accept this task, you are presented to the sociologists, who have determined several variables affecting human behaviour towards catastrophic situations. These variables are shown in Table 1.

Table 1: Variables affecting human behaviour towards catastrophic situations.

Variable	Type	Range
Stamina(s)	`int`	[0-100]
Location(p)	`int`	[1-500]
Ability(a)	`float`	[0-1]
Stress(t)	`float`	[0-1]

These variables are employed in the mathematical model defined in Equation 1.

$$\%survival(x) = \alpha \times stamina + (\beta - \frac{location}{2500}) + \gamma \times ability + (\rho - \frac{stress}{4}) + \lambda. \quad (1)$$

where x is an individual, $\alpha = \frac{7}{2000}$, $\beta = \frac{1}{4}$, $\gamma = 0.2000$, $\rho = 0.2500$ and λ is a random factor called *the luck factor* (l). While this luck factor cannot be determined in practise, the sociologists require you to allow them to decide whether to use it or not during a simulation. The luck factor is calculated as a random number in the ranges $[-0.1500 : 0.1500]$ (using four decimal digits).

2 Your first deliverable

You have to implement a **C++ program** that calculates the rate of survival of a person, given the input of the aforementioned variables. This deliverable should be implemented using at least one header (`.h`) file and an implementation (`.cpp`) file. The program must operate via command line, as indicated in the following paragraphs.

To run your program, you must be able to input parameters using flags `s`,`p`,`a`,`t` and `l`, as indicated below. An example run is the following:

```
./survivalrate -s 75 -p 278 -a 0.4500 -t 0.6500 -l
```

Each parameter introduced while running the program corresponds to each variable defined on Table 1 (identify the letters for each parameter inside the brackets) and the luck factor defined by the parameter `-l`. While the four variables on the table (other than `-l`) are mandatory to each program run, the luck factor is optional. The program must validate that the inputs are valid according to the ranges stated on Table 1. Any other factor missing should cause an error and the program must display which parameter is missing to the tester. The output of the program must be a number between 0 and 100%. If the luck factor is applied and the survival rate is greater than 100%, this value must be displayed as 100%.

3 Output Sample

The output of your program should display all the parameter values in the form of a table. However, if the luck factor is enabled, the program must run ten times and calculate an average with the standard deviation between the measures performed.

```
These are the values for the parameters of the simulated survivor:
Stamina [0--100]      75
Location [1--500]     278
Ability [0--1]        0.45
Stress [0--1]         0.65
Luck factor value (enabled)

The survival rate of this person is 53.6417%(±8.5049%)
```

4 General Requirements

- Your deliverable must be submitted using Blackboard. For a guide on how to upload assignments, refer to the next section.
- After the due date, you will have 5 *business* days to submit your assignment as a *late submission.* Late submissions will incur a penalty of 10% per day. After these five days, Blackboard will be closed and you will lose ALL the assignment marks.

- ONLY include the source code of your program, along with its corresponding `Makefile`. No object nor executable files should be required, and use the executable name specified in the assignment instructions above (i.e. `survivalrate`), otherwise you will automatically be deducted 1 mark.

- Stick to proper coding styles and OO design principles using C++. If you have question on what coding style means, please refer to `http://geosoft.no/development/cppstyle.html/` for a coding style example. Marks will **NOT** be awarded if your solution uses C standards rather than C++.

- It is mandatory to have at least a header file for the class modelled and at least one `.cpp` file for implementing the methods defined in the header and the `main()` method. Also, you must properly comment your code. As a minimum requirement, your classes, methods and attributes should be properly described in the code.

- Your assignment must primarily run on the Yallara server, using `gcc` to compile and should not return any warning or error after compiling. Failure in these requirements will automatically deduct 20% out of your assignment marks.

- If you made any assumption, used specific parameters and/or flags for compilation or anything that you consider would affect the process of marking your assignment, please include a README file with your assignment describing in detail your concerns. If such a file is not included and your assignment cannot be run, you will NOT have a chance to ask for remarking. Also, remember that the markers must NOT edit any line of code in your files to run your program. So avoid hard-coding paths (although for this assignment they are not required).

5 Submission

If you have not done it before, please refer to `http://www.youtube.com/watch?v=59iYdvx4Wyk\&feature=related` for a video guide on how to upload assignments via Blackboard.

6 Further Information

Further enquiries about this assignment should be raised in appropriate discussion board in Blackboard.

7 Plagiarism

The assignment that you submit must be your own work. No marks will be given for pieces of code that have not been created by you. Plagiarism is a serious offence at RMIT University. This includes direct copy of code from the Internet, other students or resources without a proper reference. While working in groups for an assignment is valid and encouraged, the submission of similar files is considered plagiarism as well. Having similar ideas is not a justification for not creating your own assignment files. Before marking, the markers will run plagiarism-detection tools. Plagiarised assignments will incur in penalties that range from assingment invalidation, to automatic fail of the course or even being expelled from the School if incurring for the second time. Be careful!