Assignment 2

Deadline:	Hand in by 5pm on Monday 9th September 2013 (after mid-semester break)
Evaluation:	20 marks – which is 20% of your final grade
***Late Submission**:*	1 Mark off per day late
Work	This assignment is to be done **individually** – your submission may be checked for plagarism against other assignments and against Internet repositories. If you adapt material from the Internet acknowledge your source.
Purpose:	To get you thinking about how to use generics, templates and other OO mechanisms for practical software implementation and packaging problems.

Problem to solve:

Write and comment/document an appropriate Random Number Generation Library/Package in C++, Java or Python, that supports at least **TWO DIFFERENT** fundamental generator algorithms (that can be chosen by the user at run time) but which supplies the same user interface while maximising implementation code reuse.

Requirements:

The user's program should be able to call your routines to obtain:

1) *Uniform-distributed random numbers - **uniform()***
2) *Gaussian-distributed random numbers - **gaussian()** with mean 0 and variance 1.0*
3) *Flat distributed random integers - **index(int max_not_inclusive)***
4) *Supply a test driver program that generates (say) 1,000,000 deviates and computes and prints out the average and standard deviation.*

There should be some mechanism of your choosing that allows the user to initialise the generator sequence; saves and reloads the generator state; and that allows choice of which fundamental algorithm to use.

*You must write a **report** on your random number generators, briefly describe each RNG, discuss the advantages and disadvantages of each generator and analyse them. This analysis should include performance (time taken to generate numbers) and the quality of the random numbers each produces. The methods used to analyse the random number sequences are up to you but should include at least one of the diehard tests.*

Hand-in: Submit your **program and documentation** (a zip file is acceptable) by email to d.p.playne@massey.ac.nz
Marks will be allocated for: correctness, fitness of purpose, utility, style, use of sensible **comments and program documentation**, and general style! Good comments will help me to award you marks even if your code is not quite perfect.

Additional Notes on the Technical Specification and Hints:

Choose which language to do this assignment in – either C++ or Java or Python

Think what software mechanisms you might use to maximize code reuse of your system.

Choose at least two different random number generator algorithms to implement – Lagged Fibonnacci, Linear Congruential Generation; Mersenne Twistor; or …

How will you reuse the code for making Gaussian distribution (eg use the Box-Muller algorithm)?

What interface choices will you give the user to initialize your generator? Save its state to file? Restart it from file? Start it from system time and date…?

How will you document your system and design choices?

How will you optimise your design for speed and/or ease of use?

How fast does your system generate random numbers?

For extra credit implement a Binomial distribution routine **binomial(int n, double p)**

If you have any questions about this assignment, please ask the lecturer.