CSCI251/CSCI851 Autumn-2020
Advanced Programming (**LT4**)

Lecture Tutorial 4

Outline

- From the lab:
- Assignment.
- Proceeding procedurally.

From the lab:

- The expectation is that you listen to the relevant lecture material prior to the time of the lecture Q&A session, and therefore prior to the lab.
- Debugging:
  - Remember: Logical errors, not just compile time.
    - This means that just because your program compiles, doesn't mean you have finished debugging.

- Difference between Debug-A, Debug-B.
  - There is no difference between what the two programs are supposed to do.
  - Variable names don’t matter to the compiler.
  - They make a big difference to the readability though.

- Indentation, use it consistently.

Binary file format…

- To clearly see the difference between outputting in binary and not, you need to more directly access the I/O streams, using read, write, get, put.
- Try using put to output a single character from a stored number.
  - You won't get a digit …
- Try using write …

```
outFile.write((char*)&forOutput,sizeof(forOutput));
```

```
float number = 111.11;
ofstream outFile;

outFile.open("Store_bin",ios::out|ios::binary);
cout << sizeof(number) << endl;

outFile << number << endl;
outFile.write((char*)&number,sizeof(number));
outFile.close();
```

```
111.11
R8<DE>B
```

- This is a different way of outputting and directly access the I/O.
- The number takes up 4 bytes in memory and on the second line is written in that way it is represented in memory.
  - Note that memory organisation is system dependent.

New Random…

```
$ ./a.out 1000
Generating values :
Outputting distributions :
Value : 0 --- Frequency :79
Value : 1 --- Frequency :112
Value : 2 --- Frequency :97
Value : 3 --- Frequency :101
Value : 4 --- Frequency :104
Value : 5 --- Frequency :106
Value : 6 --- Frequency :107
Value : 7 --- Frequency :96
Value : 8 --- Frequency :99
Value : 9 --- Frequency :99
```

- Output something like that illustrated above.
- The distribution here is the count of digits.
- Note the command line argument of 1000, use

```
int main( int argc, char *argv[]){ }
```

Random

- Means and standard deviations …
- A normal distribution is what you need…

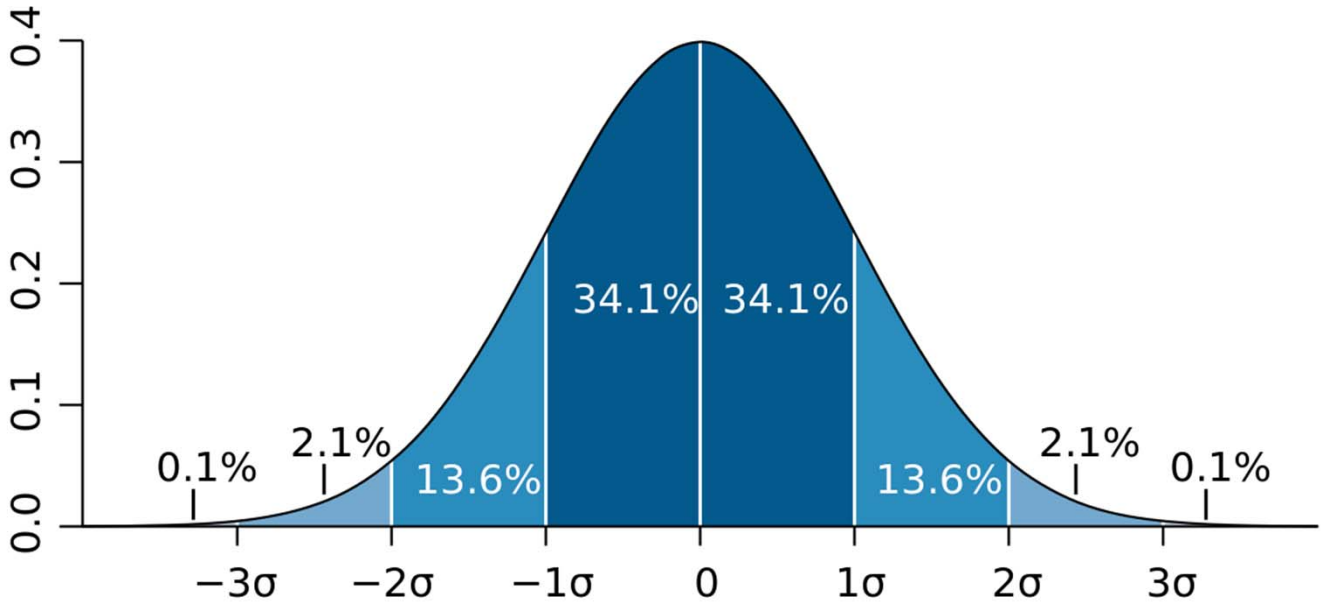

Diagram from
https://en.wikipedia.org/wiki/Standard_deviation