C++ for Gamebryo: Review for the Introductory/Beginners’ Track

References:
- Ivor Horton’s Beginning Visual C++ 2008 (available on-line) and
- Programming (Principles and Practice Using C++) by Bjarne Stroustrup.

Lecture One: Getting Started

Programming is the art of expressing solutions to problems so that a computer can execute those solutions. Why would you want to learn programming? Our civilization runs on software. Like mathematics, programming – when done well – is a valuable intellectual exercise that sharpens our ability to think. However, thanks to feedback from the computer, programming is more concrete than most forms of math, and therefore accessible to more people. It is a way to reach out and change the world – ideally for the better. Finally, programming can be great fun. Now, let’s get back to the down-to-earth business of learning the technical skills needed to program.

```cpp
#include <iostream>
using namespace std;

int main()
{
    cout << "Hello, and how are you doing?\n";
    return 0;
}
```

This program simply communicates with the user (one-way).

```cpp
#include <iostream>
#include <string>
using namespace std;

int main()
{
    cout << "Hello, what's your name? ";
    string name;
    cin >> name;
    cout << "Well, " << name << " how old are you? ";
    int age;
    cin >> age;
    cout << name << " next year you will be " << (age + 1) << endl;
    return 0;
}
```

This program communicates both ways. But careful how you type your name.

Types of variables we will encounter most: int, double, char, string, bool. At least for the time being.
```cpp
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string word = "";
    int count = 0;
    while (cin >> word) {
        count += 1;
        cout << count << ". " << word << endl;
    }
}
```

This program reads from the keyboard, counts and reports words individually. Notice that the reporting is mixed with the input (between two user newlines). To end the program type Ctrl-Z, alone, on a line.

```cpp
#include <iostream>
#include <cmath>
using namespace std;

int main()
{
    double x, y, a, b, distance;
    cout << "Enter the four coordinates separated by spaces: ";
    cin >> x >> y >> a >> b;
    distance = sqrt(pow(x - a, 2) + pow(y - b, 2));
    cout << "The distance between (" << x << ", " << y << ") and (" << a << ", " << b << ") is: " << distance << endl;
}
```

This program calculates the distance between two points in the plane. Coordinates for the two points are first entered: the x and y of the first point followed by the x and y of the second point.

```cpp
#include <iostream>
using namespace std;

int main() {
    int size;
    cout << "What size: ";
    cin >> size;
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++)
            cout << "* "; // [1]
        cout << endl;
    }
}
```

This program prints a square block of stars. The size is specified by the user.

```cpp
if (i == j || i + j == size-1)
    cout << "+ ";
else
    cout << " ";
```

Replace the highlighted code above with this if statement to obtain a scalable X.
Delegating responsibilities to a function. Note how the declaration must precede the invocation.

Sorting a sequence of numbers in descending order.
Write a program that translates a number between 0 and 4 into the closest letter grade. Letter grades are A, B, C, D and F and their numeric values are 4, 3, 2, 1 and 0. A plus or a minus add or subtract 0.3 respectively so, for example, B- is 2.7, C+ is 2.3 and so on. There is no F+ but D- is still 0.7 as expected. Thus, if we convert the number 2.8 the closest letter grade is B- since 2.8 is between 2.7 and 3.0 but 2.7 is closest. Break ties in favor of the better grade; for example 1.85 should be a C (can you see why? let me know if this is unclear.)

```cpp
#include <iostream>
using namespace std;
int main() {
    double grade;
    cin >> grade;
    if (grade > 4) cout << "Number too big."
        " << endl;
    else if (grade >= 3.85) cout << " A "
        " << endl;
    else if (grade >= 3.50) cout << " A-
        " << endl;
    else if (grade >= 3.15) cout << " B+
        " << endl;
    else if (grade >= 2.85) cout << " B
        " << endl;
    else if (grade >= 2.50) cout << " B-
        " << endl;
    else if (grade >= 2.15) cout << " C+
        " << endl;
    else if (grade >= 1.85) cout << " C
        " << endl;
    else if (grade >= 1.50) cout << " C-
        " << endl;
    else if (grade >= 1.15) cout << " D+
        " << endl;
    else if (grade >= 0.85) cout << " D
        " << endl;
    else if (grade >= 0.35) cout << " D-
        " << endl;
    else if (grade >= 0.00) cout << " F
        " << endl;
    else cout << "Number too small.";
}
```

Here’s this program in action:

```
C:\Windows\system32\cmd.exe
Enter grade: 3.2
B+
Press any key to continue . . .
```

Every time you call this program it converts one number into the closest letter grade.