Programming Review

Simple C++ Program

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

int main()
{
    //Note
    cout << "Hello World" << endl;

    return 0;
}</pre>
```

Variables

```
#include <iostream>
using namespace std;
int main()
    double gpa;
    gpa = 4.0;
    //Note
    cout << "GPA: " << gpa << "\n";</pre>
    return 0;
```

Formatting

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
    double gpa;
    gpa = 4.0;
    //Note
    cout << fixed << showpoint << setprecision(2);</pre>
    cout << "GPA: " << gpa << "\n";
    return 0;
```

Conditionals

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
    double gpa;
    gpa = 4.0;
    //Note
    cout << fixed << showpoint << setprecision(2);</pre>
    cout << "GPA: " << gpa << "\n";
    cout << "Letter Grade is "</pre>
    if (gpa > 3.67)
        cout << "an A";
    else
        cout << "not A";
    return 0;
```

Control Structures

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
    int times;
    double gpa;
    times = 3;
    qpa = 4.0;
    //Note
    cout << fixed << showpoint << setprecision(2);</pre>
    cout << "GPA: " << gpa << "\n";
    cout << "Letter Grade is ";</pre>
    if (gpa > 3.67)
        cout << "an A";
    else
        cout << "not A";
    for (int i = 0; i < times; i++)
        cout << "!";
    return 0;
```

Functions

```
#include <iostream>
#include <iomanip>
                                                       double calculateGPA()
using namespace std;
                                                            return 4.0;
int main()
    int times;
    double gpa;
    times = 3;
    gpa = calculateGPA();
    //Note
    cout << fixed << showpoint << setprecision(2);</pre>
    cout << "GPA: " << gpa << "\n";
    cout << "Letter Grade is ";</pre>
    if (gpa > 3.67)
        cout << "an A";
    else
        cout << "not A";
    for (int i = 0; i < times; i++)
        cout << "!";
    return 0;
```

Function Overloading

```
#include <iostream>
#include <iomanip>
using namespace std;
                                            double calculateGPA()
int main()
                                                return 4.0;
    int times;
    double gpa;
                                            double calculateGPA (double g1)
    times = 3;
                                                return g1;
    gpa = calculateGPA();
    //Note
    cout << fixed << showpoint << setprecision(2);</pre>
    cout << "GPA: " << qpa << "\n";
                                            double calculateGPA (double g1, double g2)
    cout << "Letter Grade is ";</pre>
                                                return (q1 + q2) / 2.0;
    if (gpa > 3.67)
        cout << "an A";
    else
        cout << "not A";
    for (int i = 0; i < times; i++)
        cout << "!";
    return 0;
```

Default Function Parameters

```
#include <iostream>
                                   double calculateGPA (double g1 = 4.0)
#include <iomanip>
using namespace std;
                                        return q1;
int main()
    int times;
                                   double calculateGPA (double q1, double q2)
    double gpa;
                                        return (q1 + q2) / 2.0;
    times = 3;
    gpa = calculateGPA();
    //Note
    cout << fixed << showpoint << setprecision(2);</pre>
    cout << "GPA: " << qpa << "\n";
    cout << "Letter Grade is ";</pre>
    if (qpa > 3.67)
        cout << "an A";
    else
        cout << "not A";
    for (int i = 0; i < times; i++)
        cout << "!";
    return 0;
```

Pass By Reference

```
#include <iostream>
                                   double calculateGPA(int& times, double g1 = 4.0)
#include <iomanip>
                                   {
                                       times = 1;
using namespace std;
                                       return q1;
                                   }
int main()
                                   double calculateGPA (int times, double g1, double g2)
    int times;
    double gpa;
                                       times = 2;
                                       return (g1 + g2) / 2;
    gpa = calculateGPA(&times); ;
    //Note
    cout << fixed << showpoint << setprecision(2);</pre>
    cout << "GPA: " << gpa << "\n";
    cout << "Letter Grade is ";</pre>
    if (qpa > 3.67)
        cout << "an A";
    else
        cout << "not A";
    for (int i = 0; i < times; i++)
        cout << "!";
    return 0;
```

Random Number Generation

```
#include <iostream>
                                   double calculateGPA(int& times, double g1 = 4.0)
#include <iomanip>
#include <cstdlib>
                                       times = rand() % 10;
#include <ctime>
                                       return q1;
using namespace std;
int main()
                                   double calculateGPA(int times, double g1, double g2)
                                       times = (rand() % 10) + 1;
    int times;
                                       return (q1 + q2) / 2;
    double gpa;
    srand(time(0));
    gpa = calculateGPA(&times);
    //Note
    cout << fixed << showpoint << setprecision(2);</pre>
    cout << "GPA: " << gpa << "\n";
    cout << "Letter Grade is ";</pre>
    if (qpa > 3.67)
        cout << "an A";
    else
        cout << "not A";
    for (int i = 0; i < times; i++)
        cout << "!";
    return 0:
```

#include <iostream> File I/O #include <iomanip> #include <cstdlib> #include <ctime> double calculateGPA(int& times, double g1 = 4.0) #include <fstream> { using namespace std; times = rand() % 10;return q1; int main() int times; double calculateGPA(int times, double g1, double g2) double gpa; ofstream output; times = (rand() % 10) + 1;return (q1 + q2) / 2;srand(time(0)); } gpa = calculateGPA(×); output.open("gpa.txt"); //Note output << fixed << showpoint << setprecision(2);</pre> output << "GPA: " << gpa << "\n"; output << "Letter Grade is ";</pre> if (qpa > 3.67)output << "an A";</pre> else. output << "not A";</pre> for (int i = 0; i < times; i++) output << "!"; output.close(); return 0;

#include <iostream> #include <iomanip> #include <cstdlib> #include <ctime> #include <fstream> using namespace std; int main() int times; double gpa; ofstream output; srand(time(0)); gpa = calculateGPA(×); output.open("gpa.txt"); //Note output << fixed << showpoint << setprecision(2); output << "GPA: " << gpa << "\n"; output << "Letter Grade is "; if (qpa > 3.67)output << "an A"; else output << "not A"; for (int i = 0; i < times; i++) output << "!"; output.close(); return 0;

Arrays

```
double calculateGPA(int& times, double g1 = 4.0)
    times = rand() % 10;
   return q1;
double calculateGPA(int times, double g1, double g2)
    times = (rand() % 10) + 1;
   return (q1 + q2) / 2;
double calculateGPA (double grades [], int number)
    double ret = 0.0;
   for (int i = 0; i < number; i++)
        ret += grades[i];
   ret /= number;
   return ret;
}
```

Pointers and Memory Allocation

#include <iostream>

#include <iomanip>
#include <cstdlib>

```
#include <ctime>
#include <fstream>
using namespace std;
                                          double calculateGPA(int* times, double g1 = 4.0)
int main()
                                               *times = rand() % 10;
                                              return q1;
   int *times;
   double gpa;
   ofstream output;
                                          double calculateGPA(int times, double g1, double g2)
   srand(time(0));
   times = new int;
                                               times = (rand() % 10) + 1;
   gpa = calculateGPA(times);
                                              return (q1 + q2) / 2;
   output.open("gpa.txt");
    //Note
   output << fixed << showpoint << setprecision(2);
   output << "GPA: " << gpa << "\n";
   output << "Letter Grade is ";
                                          double calculateGPA(double grades[], int number)
   if (qpa > 3.67)
       output << "an A";
                                               double ret = 0.0;
    else
                                              for (int i = 0; i < number; i++)
       output << "not A";
                                                   ret += grades[i];
    for (int i = 0; i < *times; i++)
                                              ret /= number;
        output << "!";
                                              return ret;
   output.close();
   delete times;
   return 0:
```

Need More?

- Professor Myers' COP3014 lecture notes:
 - http://www.cs.fsu.edu/~myers/c+ +/notes/index3014.html