Final Review

Lecture 19

Format

- The exam will be 6-7 problems, with some problems having multiple sub-questions
- You are allowed a single 8.5x11" piece of paper with whatever notes you want on it
  - Can be handwritten or computer printed
  - You may use both the front and back
- No calculators, books, laptops, phones, or anything besides your single page of notes may be used
- All responses must be made in **pen**

Kinds of Questions to Expect

- Explain a program or part of a program
- Translate between "normal" math expressions and their C++ equivalents
- Write your own code
- Fix incorrect code / find bugs in code
- Fill in the blank (in a program)
- Short answer

Content

Everything we've covered so far in the semester, including:

- Input and output (**`cin`** and **`cout`**)
- **`if`**-**`else`**, **`switch`** statements
- **`for`**, **`while`**, and **`do`**-**`while`** loops
- Functions, including call by value, call by reference, and array arguments
- File I/O
- Arrays
- Classes

Review Exercises

- The following slides contain exercises that will help you prepare for the exam
- These exercises are all about writing code to help remind you of the things we've done this semester
- Refer back to the exam 1, 2, and 3 review slides (and your actual exams) if you need a reminder of the style of questions

Exercise

Write a program that computes the total cost of buying Cheetos in bulk. The user enters the number of bags purchased, and the program computes and outputs the total cost using these rules:

- Less than 10 bags purchased, cost is $2/bag
- Between 10 and 20 bags, cost is $1.50/bag
- More than 20 bags, cost is $1/bag

Answer

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

int main()
{
	double cost;
	int num_items;
	cout << "Enter the number of bags purchased: ";
	cin >> num_items;
	if ( num_items < 10 )
	{
		cost = 2.00 * num_items;
	}
	else if ( num_items <= 20 )
	{
		cost = 1.50 * num_items;
	}
	else
	{
		cost = 1.00 * num_items;
	}

	cout.setf( ios::fixed );
	cout.precision( 2 );
	cout << "Total cost: $" << cost << endl;

	return 0;
}
```

Exercise

Write a program that reads 1000 values from the user, increments them all by 5, and prints out the new values. The program should read all 1000 values before printing out the 1000 incremented values.

Answer

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

int main()
{
    int i;
    double arr[1000];
    cout << "Enter 1000 values:" << endl;
    for ( i=0; i<1000; i++ )
    {
        cin >> arr[i];
    }
    for ( i=0; i<1000; i++ )
    {
        arr[i] = arr[i] + 5;
        cout << arr[i] << endl;
    }
    return 0;
}
```

Exercise

Write a program that reads characters from the user until a 'q' is entered and prints them back immediately to the screen. If the character is lower case, convert it to upper case before outputting it. Do not use any libraries other than iostream.

Answer

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

int main()
{
     char next;
     cout << "Enter characters, stopping with 'q': " << endl;
     cin >> next;

     while ( next != 'q' )
     {
          if ( next >= 'a' && next <= 'z' )
          {
               next = next + 'A' - 'a';
          }

          cout << next;
          cin >> next;
     }
     return 0;
}
```

Exercise

Write a program which reads grades from a file named grades.txt. Assume there are at most 30 grades in the file. Compute the average of the grades and then print out how far away each score is from the average (the difference of the score and the average).

Answer

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

int main()
{
     int index;
     double grades[30];
     double sum, avg;
     ifstream ifs;

     ifs.open( "grades.txt" );
     if ( ifs.fail() )
     {
          cout << "Failed to open!"
               << endl;
          return 1;
     }

     index = 0;
     sum = 0;
     while ( ifs >> grades[index] )
     {
          sum = sum + grades[index];
          index++;
     }

     avg = sum / index;
     cout << "Grade differences from the average "
          << avg << ":" << endl;
     for ( int i=0; i<index; i++ )
     {
          cout << grades[i] - avg << endl;
     }
     ifs.close();
     return 0;
}
```

Exercise

Write a function named **`max_and_min`** that is passed an array of integers and returns the maximum value and minimum value found in the array.

Answer

```
void max_and_min(int a[], int size, int& max, int& min)
{
    max = a[0];
    min = a[0];
    for ( int i=1; i<size; i++ )
    {
        if ( max < a[i] )
        {
            max = a[i];
        }
        if ( min > a[i] )
        {
            min = a[i];
        }
    }
}
```

Exercise

Write a function named **`print_vals`** that is passed an already opened output file stream and writes the odd numbers from 100 to 999 to the file stream. The function should write five of the numbers to each line in the file.

Answer

```
void print_vals(ofstream& ofs)
{
    int num_on_line = 0;
    for ( int i=101; i<1000; i=i+2 )
    {
        ofs << i << " ";
        num_on_line++;
        if ( num_on_line == 5 )
        {
            ofs << endl;
            num_on_line = 0;
        }
    }
}
```

Exercise

Write a class that represents a book. Every book should have a name, number of pages, and year published. Include an appropriate constructor and an **`output`** function that prints the information about the book. Write a **`main`** function to test the class.

Answer

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

class book
{
public:
     book(string title, int pages, int published)
     {
          name = title;
          num_pages = pages;
          year = published;
     }
     void output()
     {
          cout << name << " has " << num_pages
               << " pages and was published in "
               << year << endl;
     }
private:
     string name;
     int num_pages;
     int year;
};
```

```
int main()
{
     book hobbit( "The Hobbit", 310, 1937 );
     book irobot( "I, Robot", 272, 1950 );
     hobbit.output();
     irobot.output();
     return 0;
}
```

Exercise

Write a class that represents a planet. Every planet has a radius and a distance from the sun. Include an appropriate constructor, a function that lets you set the distance separately, and a function that returns the volume of the planet (volume of a sphere is $\frac{4\pi r^3}{3}$). Write a **`main`** function to test the class.

Answer

```
#include <iostream>
using namespace std;
class planet
{
public:
     planet(double rad, double dist)
     {
          radius = rad;
          distance = dist;
     }
     void set_distance(double dist)
     {
          distance = dist;
     }
     double volume()
     {
          const double PI = 3.14159;
          return ( 4.0/3 ) * PI *
                  radius * radius * radius;
     }
private:
     double radius;
     double distance;
};
```

```
int main()
{
     planet earth( 3959, 93000000 );
     earth.set_distance( 92955807.27 );
     cout << "Earth's volume is ”
          << earth.volume()
          << " cubic miles." << endl;
     return 0;
}
```

Wrap Up

- We've covered all of the basics of C++ programming this semester
- Next semester you will learn about multi-dimensional arrays, structs, recursion, dynamic memory allocation, and more about classes and inheritance
- We've been using C++, but as you'll see most of the basics are the same in other languages as well and the concepts apply much more broadly (a loop is a loop)