Name: ______________________________ SID: ____________________

COMP 2355, Winter 2013
Midterm Exam
February 7, 2012

READ ALL INSTRUCTIONS BEFORE TAKING THIS EXAM!

This midterm is worth 20% of your course grade. Be careful to ***read all instructions*** *for the exam before you begin. A small number of points will be deducted for not following instructions correctly.*

The exam should be written in black or blue ink. All answers should be written inside the exam booklet. No other materials are allowed during the exam. If you are unsure what a question is asking, write your assumptions. If there are more general clarifications, ask the proctor of the exam.

Write your name and SID on every page of the exam.

This exam should be entirely your own work. Students caught cheating may receive a 0 on the midterm and/or be subject to academic discipline. Additionally, it is your responsibility to make sure that other students cannot copy your work.

Part 1	__________	[of 20]
Part 2	__________	[of 40]
Part 3	__________	[of 40]
Total	__________	[of 100]
Bonus	__________	[of 4]

1. Short answer question [Do 4 of 5] [5 points each]

a) What is a stack overflow and how is one caused?

b) Suppose the struct Foo is already defined. Show how to allocate Foo on the stack and the heap.

c) What is the difference between `delete` and `delete[]`?

d) A colleague suggests that you should read a file in reverse (that is, from the end backwards to the start). Is this a good idea, a bad idea, or it doesn't matter? Why?

e) What happens if you index past the end of an array and read a value? What if you write a value?

2. Analysis questions [Do 4 of 5]: **[10 points each / 40 points total]**

a) What is the result of the expression `x^(~x)`?

What does this function do?

```
uint64_t function(uint64_t value, int cnt)
{
      uint64_t newValue = value>>cnt;
      newValue |= (value<<(64-cnt));
      return newValue;
}
```

Name: ______________________________ SID: ____________________

b) Add the appropriate lines in the function below to allocate an array of 10 items.

```
int *GetSequence(int from, int to)
{
      // perform error check







      // allocate array named resultArray




      for (int x = from; x < to; x++)
      {
            resultArray[x-from] = x;
      }

      return resultArray;
}
```

c) Take the two functions defined below and re-write them as a single function can replace both of the original two functions.

```
int SumSequence(int high, int low)
{
      int result = 0;
      for (int x = low; x < high; x++)
      {
            result += x;
      }
      return result;
}
```

```
int SumSequence(int maxValue)
{
      int result = 0;
      for (int x = 0; x < maxValue; x++)
      {
            result += x;
      }
      return result;
}
```

Name: ______________________________ SID: ____________________

d) The code below was recently refactored into three separate files. Find and fix 5 of the errors (there are more than 5) in the new code, and briefly say why these are errors. (Spaces are added uniformly to each files and do not indicate where errors may be found.)

main.cpp	factorial.h	factorial.cpp

#include <factorial.h>
int main(int, const char **)
{
  uint64_t value = 10;
  cout << factorial(value);
  return 0;
}
	namespace Util {
	  uint64_t factorial(int x)
	  int getMaxFactorial()
	  {
	    return 20;
	  }
	}
		namespace Util {
		uint64_t factorial(int x)
		{
		  assert(x <=
		          getMaxFactorial());
		  if (x < 1)
		    return 1;
		  return x*factorial(x-1);
		}
		}

e) Fill in the sample calls for each function.

```
struct myStruct {
      int x, y;
};

void InitializeStruct1(myStruct &initMe)
{
      initMe.x = 0;
      initMe.y = 0;
}

void InitializeStruct2(myStruct *initMe)
{
      initMe->x = 0;
      initMe->y = 0;
}

int main(int, const char **)
{
      myStruct a;
      myStruct *b = new myStruct;

      // pass in a
      InitalizeStruct1(                                   );

      // pass in b
      InitalizeStruct1(                                   );

      // pass in a
      InitalizeStruct2(                                   );

      // pass in b
      InitalizeStruct2(                                   );
}
```

Name: ________________________________ SID: ______________________

3. Synthesis questions [Do 4 of 5] [10 points each]

a) Write a function `SetBit` which will take two parameters, a `uint64_t` and an `int`. The function should set the *n*th bit in the bits. (The bit should be changed in the calling function after the function returns.)

b) Write the function DeepCopy which takes as an argument a linked list, and returns a (deep) copy of the list. That is, all items in the list should be copied. The list data structure is defined below.

```
struct list {
      // for this question we don’t need any items in the list
      // but in practice they might be here. Your function shouldn’t take
      // advantage of the fact that there is no other data in this struct
      list *next;
};

list *DeepCopy(const list *original)
{









}
```

c) Write a function FreeList which takes as an argument a linked list and frees all the memory associated with the list. (Use the same data structure from the previous question.)

```
void FreeList(list *toFree)
{








}
```

d) Write a function int Length(const char *myString) which computes and returns the length of the string passed in.

```
int Length(const char *myString)
{

}
```

e) Write a templated function *swap* which takes two values as arguments, and swaps the values. (They should be swapped in the calling function after the call completes.)

4. Bonus Questions (extra credit) **[4 points]**

a) [2 points] What is the name and author of the course textbook.

b) [1 point] Finish the sentence: One thing I like about the class so far is:

c) [1 point] Finish the sentence: One way this class could be improved is: