**CST 8219 – W20 Max. Marks: 50 Surbhi Bahri**

**BY: Aria Gomes**

**Assignment #1**

**Due Time: 23.59, Sunday March 1st, 2020 Earnings: 10% of your final grade**

***NOTE: Plan to finish a few days early to avoid last minute hardware/software holdups for which no allowance is given.***

***NOTE: The code in this assignment must be your own work. It must not be code taken from another student or written for you by someone else, even if you give a reference to the person you got it from (attribution); if it is not entirely your own work it will be treated as plagiarism and given a fail mark, or less.***

**Purpose: You are to write the code in Visual Studio 2019 wherever it is asked for.**

1. **Write a program utilizing the ‘Length()’ function that takes the length of the linked list and computes the number of elements in the list. Make sure you pass the list by passing the Head**

**Pointer and iterate over the same using a Local Pointer. (10Marks)**

Check option 5 in code sample, Report Length. (int Length() & void ReportLength())

1. **Consider a CopyList() function that would take a new list over the original and returns the complete copy of the same. It is mandatory to use 3 pointers; two of them should keep track of**

**the new list and the other has to iterate over the original list. (10Marks)**

Check option 6 in code sample, Copy List. (Animation\* CopyList())

**3) Variables of type void\* can point to any type of variable. Would a type void[] make sense?**

**Explain. (2Marks)**

No void[] does not make sense because a void pointer is used to point to another data type that is not void where as an array of type void is setting each element inside the array to void which is not allowed.

**4) Differentiate between Class, Struct and Union. Which to use and when? Instantiate.**

**(6Marks)**

Each one stores memory differently for example a union will allocate memory for the largest datatype inside of it where as a struct will allocate memory for the sum of each datatype inside. If a union’s data member’s value is altered others will be too. A Class is extensible program-code-template where all objects created will be stored on the heap because class is a reference type. You would use a Union when you want to figure out which data type, object or class is being accessed. You would use a structure when you need many data members that need to be accessed/modified. A class should be used to separate pieces of code that need to be reused/reassessed or separated. They would look like this when instantiated.

class [class name]  
{  
…  
}

struct [structure name]

{

member definition;

…

} alias;

union [union name]  
{  
member definition;  
…  
}

**5) Given the following Class Definitions:**

**Class A**

**{**

**public:**

**A() { }**

**A(const A&) { }**

**~A() { }**

**A& operator = (const A&)**

**{ }**

**};**

**Class B**

**{**

**private:**

**A a[2];**

**public:**

**B() { };**

**B (const B&)**

**~B() { }**

**{ }**

**B& operator = (const B& b)**

**{**

**A :: operator = (b);**

**}};**

**How many times and in what order the constructors and destructors of A and B are executed if the following function is called:**

**B fuse (B b)**

**{**

**return b;**

**}**

**………….**

**………..**

**B b1, b2;**

**b1 = fuse(b2);**

|  |  |  |
| --- | --- | --- |
|  | **Feel free to add print lines to each of the constructors, destructors and assignment** | |
|  | **operators. Attach a complete program.**  Could Not solve unfortunately because the line A::operator = (b); gave a compilation error. ☹ | **(7Marks** |

**CST 8219 – W20 Max. Marks: 50 Surbhi Bahri**

|  |  |  |
| --- | --- | --- |
| **6)** | **Why would one use pointers to member functions? How it differentiates from regular pointers to** | |
|  | **functions?**  A pointer to a member function is useful because member functions contain the “this” function which is very useful to the programmer where as using “this” on a normal function would use the scope of the entire class. The member function needs to be accessed using “::” and member function pointers need “::\*”. Where Regular pointers to functions need “pointer\_name = function\_name;” | **(4Marks)** |
| **7)** | **How can one allocate multi-dimensional arrays dynamically?** | **(2Marks)** |
| **8)** | To allocate a multi-dimensional array dynamically one would need to create a double pointer of a datatype pointer dynamically using new. Then iterate through that in a loop through for the number of rows there are, inside the loop create the same data type that you have chosen before with the loop control variable as the index for your double pointer variable using the new keyword(dynamic) once again and enter the amount of columns desired. The finished product should look something like this.  int\*\* a = new int\*[amountOfRows];  for(int i = 0; i < ammountOfRows; ++i)  a[i] = new int[ammountOfColumns];  **What is the difference between a pointer passed by value and a pointer passed by reference?** | |
|  | **Give a concrete example when the latter is needed.** | **(5Marks)** |
|  | A pointer passed by reference is a copy of the pointer’s memory address where as a pointer passed by value is the original value that is inside of the pointer. Pass by reference should be used when you will be changing the values in the function as the original values will NOT be affected as it is a copy of the original value. Passing by value is giving the memory address associated to that value meaning that if that value is changed the original value will be lost. |  |
|  |  |  |
|  |  |  |
| **9)** | **What can go wrong if a class has a copy constructor but no destructor or vice versa?** |  |
|  |  | **(2Marks)** |

If a class has no destructor you may get memory leaks on the exit of your application due to there not being any destructors to destroy your objects and classes properly

**10) Describe how and when the constructor and destructor of a class gets called implicitly.**

**(2Marks)**

The compiler will call the constructor of a method implicitly when copies of an object are created. For instance, if Frame A has been created and we create a new Frame B = A, the copy constructor will implicitly be called by the compiler. The destructor on the other hand will look for copies of the same object and destruct them with the same destructor implicitly once it has been called. So if both the Frames above are in a class called Frames and we use the ~Frames() deconstructor both Frame A & B will be destroyed.