**CEIS 420 Week 5 Homework**

**Question 1**

Describe a situation where a C# struct is preferable to a C# class.

When you are looking for a high-performance, memory-efficient item that would pass copies of itself as opposed to creating another instance of the whole object plus the inheritances. This makes a little more sense when structs are normally used for minor things like an integer, float, etc. as opposed to a large object that has multiple objects associated with it along with the inheritance and object-oriented nature of classes. You do have to be careful when using them, however as they are memory limited even though they are efficient with what they use.

**Question 2**

Review the following class definition in Java answer the following questions.

class StackClass {

private int [] stackRef;

private int maxLen,

topIndex;

public StackClass() { // A constructor

stackRef = new int [100];

maxLen = 99;

topIndex = -1;

}

public void push(int number) {

if (topIndex == maxLen)

System.out.println("Error in ­push–​­stack is full");

else stackRef[++topIndex] = number;

}

public void pop() {

if (empty())

System.out.println("Error in ­pop–​­stack is empty");

else --topIndex;

}

public int top() {

if (empty()) {

System.out.println("Error in ­top–​­stack is empty");

return 9999;

}

else

return (stackRef[topIndex]);

}

public boolean empty() {return (topIndex == -1);}

}

Questions:

1. What are the member variables?

stackRef, maxLen, topIndex

1. Does this include a destructor? Why do you need a destructor in C++ but not in Java?

Technically no, but it is included in Java automatically,

Java has a garbage collector – or – manages its own allocation and deallocation of objects in memory

1. Create a main to run this class. Paste your code below and a screenshot of it working
   1. Create a new StackClass variable.
   2. Push 2 numbers.
   3. Print out the top
   4. Pop a value
   5. Then print out the top again.

public static void main(String[] args){

StackClass stack = new StackClass();

stack.push(12);

stack.puah(24);

System.out.println(stack.top());

stack.pop();

System.out.println(stack.top());

}

A screen shot of a computer

Description automatically generated