**CS173 Intermediate Computer Science**

**Reading 12 Questions**

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_Daniel Lee\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Read Chapter 14.1 on dynamic data and watch the videos on Canvas about shallow vs. deep copying.**

**Answer the following questions:**

(1) Define dynamic data.

Dynamic data are variables created during execution of a program by means of special operations.

(2) What is the **new** command/operator and what does it do?

The new command/operator is one of the dynamic data operators in C++ and it is used to allocate a variable when a program requires an additional value.

(3) Give an example of creating a dynamically allocated integer. Give an example of dynamically allocating an array of five integers.

int \*ptr;

int \*array;

// add your code below

ptr = new int;

array = new int[5];

(4) Add statements to the same code to set the lone integer value to 10. Then use a loop to put the values 10,20,...50 in the five elements of the array.

//write your code here

\*ptr = 10;

for (int i = 0; i < 5; i++){

array[i]=10\*(i + 1);

}

(5) Correctly deallocate (delete) the lone integer and the array of integers.

// add your code statements here

delete ptr;

delete[] array;

(6) Define memory leak.

A memory leak is the loss of available memory space that occurs when dynamic data are allocated but never deallocated.

(7) What is a dangling pointer? Why is it dangerous? Write some code below to give an example of a dangling pointer.

A dangling pointer is a pointer that points to a variable that has been deallocated. The reason why it is dangerous is that if the program later dereferences the pointer variable, the result will be unpredictable.

int\* ptr = new int;

\*ptr = 16;

delete ptr; // now ptr is the dangling pointer

(8) Below is a sample of code. I have made some memory errors.

(a) Show where I have a memory leak. There may be none, one, or more than one.

Running the code line qtr = ptr + 1; causes us to lose connection with the original qtr memory and thus never gets deallocated, causing a memory leak.

There is no code that deallocates wtr, causing a memory leak.

(b) Show where I have a dangling pointer. There may be none, one or more than one.

ptr and qtr becomes a dangling pointer after running delete [] ptr;

(c) Show where I have an inaccessible object. There may be none, one, or more than one.

Running the code line qtr = ptr + 1; causes us to lose connection with the original qtr memory

int \*ptr = new int[4];

int \*qtr = new int;

int \*wtr = new int;

qtr = ptr + 1;

\*qtr = 5;

\*wtr = 10;

delete [] ptr;

\*qtr = 12;

(9) Why must we explicitly implement the default constructor, copy constructor, destructor, and assignment operator in classes with dynamic data?

A class with dynamic memory must perform deep-copy methods and this requires to explicitly implementing all the default constructor, copy constructor, destructor, and assignment operator in classes.