**CPSC 1020 Spring 2018**

**EXAM #2**

**This is a Closed Book exam. Please keep your notes and your computers closed. You have 50 minutes to complete the Exam. There is a total of 52 points. Although the points total 52 your grade will reflect 100%. EX. A total score of 51 will result in 100%, a total score of 44 will result in an 84%.**

**Question 1: (Part 1 and 2: 12 Points)[C++ Classes, dynamically allocating memory ]**

**Part 1: (Total of 9 points)**

Consider the Car class declaration on Attachment 1:

Implement the functions specified below. Assume all functions you were not asked to implement have been implemented.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Implement the regular constructor. You may use

initialization list if you wish. **(2 points)**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Implement setMake. **(2 points)**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Implement getMake. **(2 points)**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Implement printInfo – this function should print out the

Make, Model, and Year each with a space between them

and a new line at the end. You should use the stringstream class

Sample output would be:

Toyota Prius 2015

**(3 points)**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**Part 2: (3 Points)**

A: In main create an array of 5 Car pointers called carPtrs. Using C++ concepts, dynamically allocate the memory for the 5 pointers.

B: Using the printInfo function you implemented above print the information for each of the carPtrs.

int main()

{

return 0;

}

**Question 2: (6 Points). [Member Initialization list]**

**Consider the code on Attachment 2 and complete both Person constructors using the member initialization list technique.**

Person::Person() :

Person::Person(string l, string f, string e, int number, string street,string city, string state, int zip, int month, int day, int year) :

**Question 3: (2 Points) [General Object Oriented Programming]**

1. When a class “has-a” instance of another class, this is an example of \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. What type of relationship is realized by inheritance? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Question 4**: **(14 Points Total)** **This covers the next several questions.**

**Answer any questions pertaining to the Array class shown on Attachment 3. Also, Implement the destructor and operator=. You will NOT implement all of the class functions, just the destructor and operator=.**

**(1 Point)**

The function setElement(int e, int value) is considered an inline function. Explain why it is considered an inline function? (i.e., How do I know it is an inline function?)

**(1 Point)**

The compiler does something special when it sees an inline function, what is it?

**(3 points)**

**Implement the destructor**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*The destructor gives the allocated memory**

**\*back to the OS. It also decrements the static variable**

**\*remember when a destructor is invoked an instance of**

**\*the class is destroyed.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

Array::~Array()

{

}

**(6 points)**

**Implement the operator=**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*The overloaded = operator for assignment**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

Array &Array::operator=(const Array &rhs)

{

}

**(3 points)**

**Describe where the static variable should be initialized and write the code that would initialize the static variable.**

**Question 5: (3 Points) [Abstract Classes and static functions]**

Consider the implementation of the following static function:

static void Person::printInfo( ){ cout << numPersons; }

In main I would like to call the printInfo() function. Person is an abstract class and in main when I create an instance of Person to use to call the printInfo() function I get a compile error.

**Why** am I getting an error and **give** an example of how I would call printInfo() in main.

**Question 6: (3 points) [Copy constructor]**

In class we discussed three specific times a copy constructor is called. What were the three we discussed?

1.

2.

3.

**Question 7: (3 points) [Inheritance, Polymorphism]**

Consider the code on Attachment 4 as well as the following main.

int main( )

{

const int NUM\_DOGS =3;

Dog \*kennel[ ] = { new Dog(40.5),

new SheepDog(45.3, 50),

new Dog(24.7)

};

// Walk by each kennel and make the dog bark

for (int k = 0; k < NUM\_DOGS; k++){

cout << k+1 << ": ";

kennel[k]->bark( );

}

return 0;

}

Below is the output of the above main. See attachment 4 for the Classes.

1: I am a dog weighing 40.5 pounds.

2: I am a dog weighing 45.3 pounds.

3: I am a dog weighing 24.7 pounds.

This is not what I was expecting. The expected output should be as follows:

1: I am a dog weighing 40.5 pounds.

2: I am a sheepdog weighing 45.3 pounds and guarding 50 sheep.

3: I am a dog weighing 24.7 pounds.

Explain why this code is not printing what I expect and how can I fix the problem.

**Question 8: (4 Points Total) [Vectors]**

Below are several questions pertaining to vectors.

1. We know we can access an element of a vector using the bracket operator [ ], much like an array. We can also access an element using a function provided by the vector class. **Using an example, what is the function and what is the advantage of using the function rather than using [ ]? (2 Points)**
2. In **one** line of code create a vector or type double that has a size of 10 with everything initialized to 5.2. **(1 Point)**
3. Later I decided I need the vector, created in 2 above, to have 25 elements rather than 10. Use **resize** to change the size to 25 setting the values to 5.2. **(1 Point). This should only be 1 line of code.**

**Question 9: (5 Points)[Constructors/operator=]**

Consider the code on attachment 5

What is the output of this program? Be sure to pay attention to the return value of any functions that will be called.

int main()

OUTPUT:

{

Point p1(10, 15);

Point p2(5,5);

Point p3;

p3 = p2 = p1;

return 0;

}