**CPSC 1020 FALL 2017**

**EXAM #3**

**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 50 points. Although the points total 50 your grade will reflect 100%. EX. A total score of 50 will result in 100%, a total score of 25 will result in a grade of 50%.**

**Question 1: Vectors (4 points)**

In class we discussed 3 advantages vectors have over arrays. List 2 of the three we discussed.

1.

2.

**Question 2: Vectors (14 points total)**

1. Create an empty vector of type **double** called ***lions***. (1 point)
2. Create a vector of type **double** called ***tigers*** that has a size of 10 and each element to have 5.5. This should only be one line of code. (2 point)
3. Create a vector called ***bears*** with the same number of elements and values as ***tigers*** (created above). This should only be one line of code. (1 point)
4. In “**A”** above, you created an empty vector called ***lions****,* add a double to this vector with the value of 19.50. (1 point)
5. In “**B**” above, you created a vector called ***tigers*** that has 10 elements, using the function discussed in class add 15 **elements** to this vector. This should only be one line of code. (1 point)
6. Consider the vector “***bears”***  created in “**C**” above. (2 points 1 each)
   1. What will bears.pop\_back() return?
   2. What will bears.back() return?
7. Assume you have a vector of integers called ***dogs*.** Use a loop to traverse through each element of ***dogs*** adding 1 to the element. Use the element access mechanism that provides bounds checking for vectors. (3 points)
8. Assume you have a class called ***Date*** that has three variables of type int for month, day, and year. Also assume the ***Date*** class has a regular constructor that has 3 parameters that represents month, day, year. Create a vector of type ***Date*** called ***bDay*** with an initial size of 2 and instantiates the 2 instances of ***Date***. This should only be one line of code. (3 points)

**Question 3 (3 points)**

In class we discussed various types of constructors, one of which is a copy constructor.

**Briefly** explain why we should override the default copy constructor provided by C++.

**Question 4 (9 points)**

Below is a declaration for a class called Array. You are to implement 1) the copy constructor, 2) the Destructor, and 3) the operator=

class Array

{

private:

int size;

int \*ptr;

public:

Array();

Array(int aSize);

Array(const Array &obj);

~Array();

int getSize()const;

static int getArrayCount();

Array &operator=(const Array &obj);

void print(int) const;

void setElement(int e, int value) ;

};

1. **Copy constructor** implementation goes here:
2. **Destructor** implementation goes here:
3. The **operator=** implementation goes here:

**Question 5 (6 points)**

On the last page of this exam you will find the class declaration for Address, Date, and Person. Complete the 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 6 (4 points)**

**Consider the following class definition:**

class NumberArray

{

private:

double \*aPtr;

int arraySize;

public:

1: NumberArray operator=(const NumberArray &right);

2: NumberArray(const NumberArray &);

3: NumberArray(int size, double value);

~NumberArray();

void print();

void setValue(double value);

};

The class above has 3 numbered functions. Consider each line of code in the main. Label each with the number that represents the function that is called for the line of code.

int main()

{

\_\_\_\_\_\_\_ NumberArray first(3, 10.5);

\_\_\_\_\_\_\_ NumberArray second = first;

\_\_\_\_\_\_\_ NumberArray third(5, 12.4);

\_\_\_\_\_\_\_ third = first;

}

**Question 7: (6 points)**

The following program will test your knowledge of when base class and derived class constructors and destructors are called.

Consider the following code, determine and write the COMPLETE output below.

#include <iostream>

using namespace std;

**OUTPUT:**

// Base class

class BaseDemo{

public:

BaseDemo() // Constructor

{ cout << "This is the BaseDemo constructor.\n"; }

~BaseDemo() // Destructor

{ cout << "This is the BaseDemo destructor.\n"; }

};

// Derived class

class DeriDemo : public BaseDemo{

public:

DeriDemo() //Constructor

{ cout << "This is the DeriDemo constructor.\n"; }

~DeriDemo() // Destructor

{ cout << "This is the DeriDemo destructor.\n"; }

};

int main(){

cout << "We will now create a DeriDemo object.\n";

DeriDemo object;

cout << "The program is now going to end.\n";

return 0;

}

**Question 8: (2 points)**

With respect to Static Member Variables we discussed three important things about static member variables. What are they?

1.

2.

3.

**Question 9: (1 point)**

In object oriented programming the “has-a” relationship is known as: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Question 10: (1 point)**

In object oriented programming the “is-a” relationship is known as: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

class Address

{

private:

int house;

string street;

string city;

string state;

int zip;

public:

Address(int h, string str, string c, string st, int z);

};

class Date

{

private:

int month;

int day;

int year;

public:

Date();

Date(int m, int d, int y);

};

class Person

{

private:

string last;

string first;

string email;

Address address;

Date bday;

public:

Person();

Person(string l, string f, string e, int house, string street, string city, string state, int zip,

int month, int day, int year);

};