**CPSC 1020 Spring 2017**

**EXAM #2**

**This is a Closed Book exam. There is 105 total points. 5 points extra credit**

**Question 1: Classes (19 points)**

**Consider the following class declaration and fill in the code below:**

class Time

{

private:

int hour;

int minute;

int second;

public:

Time( );

Time(int, int, int);

void setHour(int hour);

void setMinute(int minute);

void setSecond(int second);

int getHour() const;

int getMinute() const;

int getSecond() const

void printTime();

};

**Using the class definition above, implement the following functions.**

**QUESTION 1**

1. Using initialization list, write the default constructor. **5 points**

Time::Time( )

{

}

2. Write the regular constructor. **You should make sure the hour, minutes, and seconds are within the appropriate limits.** If they are not print a statement indicating an invalid time has been entered and exit the program. Do not use initialization list with this constructor. **DO NOT USE** **MILITARY TIME.**  **5 points**

Time::Time(int hour, int minute, int second )

{

}

3. Write the setter for hours. **2 points**

void Time::setHour(int hour )

{

}

4. Write the code to print the time in the following format hh:mm:ss (hour minute second

Ex 03:15:59 **2 points**

void Time::printInfo( )

{

}

5. When writing a class, or any program for that matter, you should write small increments of code and test. I want you to complete the main with code that will test your constructors as well as the printInfo function for the Time class. You do not need to test the set function. **5 points**

#include <iostream>

using namespace std;

#include “Time.h”

int main( )

{

return 0;

}

**Question 2: Vectors (7 points)**

Consider the following code. What is the complete output for this program?

#include <iostream>

#include <vector>

using namespace std;

int main()

{

vector<int> values;

values.push\_back(1);

values.push\_back(2);

values.push\_back(3);

cout<< “The size of values is”<< values.size() << endl;

cout << “Popping a value from the vector...\n”;

values.pop\_back();

cout << “The size of values is now” << values.size() << endl;

cout << “Popping a value from the vector...\n”;

values.pop\_back();

cout << “The size of values is now” << values.size() << endl;

cout << “Popping a value from the vector...\n”;

values.pop\_back();

cout << “The size of values is now” << values.size() << endl;

return 0;

}

**Question 3: Inline functions (6 points)**

Consider the following code:

Class A

{

public:

int add(int a, int b);

};

We discussed two ways to create an inline function. In no more than 1 sentence each, explain the two ways to make the above function called add an inline function. (2 points each)

a.

b.

**Question 4: Dynamic Memory Allocation (22 points)**

1. Create a pointer of type **int** called **arrayptr.** (2 points)
2. In C++, using the pointer from “**a”** above dynamically allocate memory for 100 integers. (5 points)
3. Write the code to set the value of the memory allocated above to consecutive odd numbers starting with 1. This is not an entire program just a loop that sets the value of memory pointed to by arrayptr. The values should be 1, 3, 5, 7, ….. etc (10 points)
4. Assume, we are now finished with the dynamically allocated memory for arrayptr and want to give the heap memory back to the OS. Write the code to give the memory set aside for the 100 integers in **“b”** above back to the OS. (5 points)

**Question 5: Dangling Pointer and Memory Leaks (4 Points)**

A pointer is dangling if it contains the address of memory that has been freed. In the notes, I listed the solution to a dangling pointer. What is the solution: (2 points)

A memory leak occurs if no-longer-needed dynamic memory is not freed. The memory is unavailable for reuse within the program. What can you do to correct this problem? (2 points)

**Question 6: (2 points)**

In class and in the notes, I used a car as an analogy to describe a specific concept. The analogy was that I do not need to know how everything works under the hood of a car in order to drive a car. Also, the details of internal components, organization, and operation of cars, microwaves, blue-ray players, etc. are kept separate from the description of how they work. In object oriented programming (OOP) this is known as \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

**Question 7: (4 points)**

There are two common programming methods used today. List the two.

a.

b.

**Question 8: (2 points)**

In class we discussed two reasons Data Hiding or Information hiding is an important concept in software development. **In one sentence**, **what was one of the reasons we discussed**.

**Question 9: (4 points)**

A class in C++ can have a member variable that is an instance of another class. This represents a “has a” relationship between classes. This relationship is referred to as \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. (2 points)

In C++, a class that is a specialization of another class represents a “is-a” relationship between classes. This is called \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. (2 points)

**Question 10: (4 points)**

In class we discussed several ways vectors are similar to arrays. List two. (2 points each)

a.

b.

**Question 11: (4 points)**

In class we discussed several advantages of vectors over arrays. List two. (2 points each)

a.

b.

**Question 12: (4 points)**

What is the output of the following code.

#include <iostream>

#include <vector>

using namespace std;

int main()

OUTPUT:

{

int i;

vector<int>test(15, -1);

cout << test.size() << endl;

test.resize(25, -1);

cout << test.size() << endl;

test.push\_back(5);

cout << test.size() << endl;

test.back();

cout << test.size() << endl;

return 0;

}

**Question 13: (5 points)**

There are two ways to access an element in a vector. One using array notation [ ] and the other is using a function provided by the C++ Standard Template Library “at( )” . There is one advantage of using the “at( )” function over the array notation. What is that advantage?

**Question 14: (5 points)**

Define a vector of type **char** of size 25 and initialize each character to the capital letter ‘Y’. This should only be one line of code. You are not allowed to use a loop or a function to do this.

**Question 15: Static Variables (6 points)**

In class we discussed three important facts in reference to Static Member Variables. In one sentence each describe the three.

1. Must be:

2. Must be:

3. Can be:

**Question 16: Inheritance (7 points)**

Consider the following classes:

Draw the inheritance hierarchy of these classes. Ignore the fact these classes don’t have any implementation.

class Shape

{

some class information

};

class Quadrilateral : public Shape

{

some class information

};

class Square: public Quadrilateral

{

some class information

};

class Rectangle: public Quadrilateral

{

some class information

}

class Curved2D : public Shape

{

some class information

}

class Circle : public Curved2D

{

some class information

};

class Ellipse : public Curved2D

{

some class information

};