**CPSC 1020 SPRING 2016**

**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 113 points ( 13 extra credit)**

**Question 1: (50 points)**

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

class CpscCourse

{

private:

string courseName;

int courseNum;

public:

/\*Constructors\*/

//Default Constructor

CpscCourse();

//Paremetized Constructor

CpscCourse(string, int);

/\*Destructors\*/

~CpscCourse();

/\*Setters\*/

void setCourseName(string);

void setCourseNum(int);

/\*Getters\*/

string getCourseName();

int getCourseNum();

/\*Member Functions (Methods)\*/

void printInfo();

};

**QUESTION 1 PART 1**

1. /\*Write the Default constructor. **5 points**\*/

CpscCourse::CpscCourse()

{

cout << “The default has been called.” << endl;

//FILL IN HERE

}

2. /\*Write the paremetized constructor **5 points**\*/

CpscCourse::CpscCourse(string cName,int cNum)

{

cout << "The Paremetized constructor has been called." << endl;

//FILL IN HERE

}

/\*This is the destructor\*/

CpscCourse::~CpscCourse()

{

cout << "the destructor is being called" << endl;

}

/\*SETTERS\*/

3. /\*Write the setter for courseName **3 points**\*/

void CpscCourse::setCourseName(string cName)

{

cout << "Setting the course name" << endl;

//FILL IN HERE

}

4. /\*Write the setter for courseNum **3 points**\*/

void CpscCourse::setCourseNum(int cNum)

{

cout << "Setting the course Number " << endl;

//FILL IN HERE

}

/\*GETTERS\*/

5. /\*Write the getter for courseName **3 points**\*/

string CpscCourse::getCourseName()

{

cout << "Getting the courseName" << endl;

//FILL IN HERE

}

6. /\*Write the getter for courseNum **3 points**\*/

int CpscCourse::getCourseNum()

{

cout << "Getting courseNum" << endl;

//FILL IN HERE

}

7. /\*Write the code to print out the name and course number of a CPSC course **5 points**\*/

void CpscCourse::printInfo()

{

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* " << endl;

//FILL IN HERE

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* " << endl;

cout << endl << endl;

}

**Question 1 Part 2**

**This is the main for the above CpscCourse class (assume the class is defined in CpscCourse.h and the implantation is in CpscCourseImp.cpp. Fill in the #include below**

#include <iostream>

#include <cassert>

#include <vector>

#include **FILL IN THIS #INCLUDE** **3 point**

using namespace std;

int main()

{ /\*THIS PART IS VERY SIMILAR TO THE VECTOR HOMEWORK YOU WERE GIVEN\*/

1. Delcare a vector size 5, type CpscCourse, variable name courses. **5 points**

string cName;

int cNum;

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

{

1. Use cin to get the **name** of the course (use the variable provided above) **2 points**

1. Using the vector you created (#1) set the course **name**. **3 points.**
2. Use cin to get the **number** of the course (use the variable provided) **2 points**
3. Using the vector you created (#1) set the course **number** the user entered. **3 points**.

1. Now print the course information by calling the print function you created above. **5 points**

}

return 0;

}

**Question 2: (10 points)**

The code below does not actually swap the values of a and b. Using C++ **PASS BY REFERENCE (NOT A POINTER)** fix the code so that it will swap the values of a and b correctly.

**Hint: this basically only involves changing two lines of code.**

#include <iostream>

using namespace std;

// function declaration

void swap(int x, int y);

int main ()

{

// local variable declaration:

int a = 100;

int b = 200;

cout << "Before swap, value of a :" << a << endl;

cout << "Before swap, value of b :" << b << endl;

/\* calling a function to swap the values.\*/

swap(a, b);

cout << "After swap, value of a :" << a << endl;

cout << "After swap, value of b :" << b << endl;

return 0;

}

// function definition to swap the values.

void swap(int x, int y)

{

int temp;

temp = x;

x = y;

y = temp;

return;

}

**Question 3: (5 points)**

In class we learned that for the most part C-style arrays are the same as C++ style of arrays. However, we did discuss one major difference. What is the difference we discussed.

**Question 4: (9 points) True/False**

True / False Unlike arrays, vectors do not store the elements in contiguous memory locations.

True / False When you declare a vector, you must set the size of the vector immediately.

True / False If you add to a vector that is already full, the vector will automatically increase its size to accommodate the value.

**Question 5: (20 points)**

a. Using **one** line of code, declare a vector of 10 doubles, called nums and initialize the doubles to 1.0.

b. Using **one** line of code declare a vector called nums2 of type double and set its values to the same values as the nums vector you created above.

c. Now add an element with the value of 5.3 to the nums vector in **‘a’** above. Hint: Remember nums is of size 10 and it currently has 10 elements each with the value of 1.0.

d. Now I want to know the current size of the vector nums. Finish the code below that will determine the size of the vector nums.

int howbig = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Question 6: (14 points)**

Consider the following code. What is the output.

// This program demonstrates the vector size,

// push\_back, and pop\_back member functions.

#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 7: (5 points)**

What does pop\_back() return?

#include <stdio.h>

int main()

{

int values[10] = {-1, 14, -24, 6, 9, 2, -3, 4, 7, 3};

char word[26] = {'H', 'a', 'v', 'e', ' ', 'a', ' ','g', 'r', 'e',

'a', 't', ' ', 'w', 'e', 'e', 'k', 'e', 'n', 'd', '!', '!', '!', '!'};

int i, n=10;

int \*pv0 = values;

printf("0. pv0 = %i\n", \*pv0);

int \*pv1 = pv0 + 3;

printf("\n1. pv1 = %i\n", \*pv1);

char \*pv2;

pv2 = word;

printf("\n2. \*pv2 = %c\n", \*pv2);

printf("\n3. \*(pv2 + 3) = %c\n\n", \*(pv2 + 3));

// output

printf("\n4. Check: ");

char \*pv4 = word;

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

printf("%c", \*pv4);

pv4 += 3;

}

printf("\n\n5. %s\n", word);

// exit

printf("\nAll done!\n");

return(0);

}