**CPSC 1020 Spring 2018**

**EXAM #1**

**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 51 points. Although the points total 51 your grade will reflect 100%. EX. A total score of 51 will result in 100%, a total score of 44 will result in an 86%.**

**Question 1: ( 9 points) [STUCTS, POINTERS TO POINTERS, and C-STYLE STRINGS]**

**Part 1: (3 points)**

Create a **C-style** struct. The struct will represent a car with three data members: the make, the model, and the year. Use typedef and call it **car\_t**. You can assume make and model will be < 20 characters each.

**Part 2: (3 points)**

Create an array of **car\_t** pointers called **carPtrs**. Using “C” concepts, dynamically allocate the memory for 10 pointers.

**Part 3: (3 points)**

Set the first element in **carPtrs** to the following values – make = Honda, model = CRV, year = 2018

**HINT:** Since this is “C” we represent strings using char arrays. You might want to use **strcpy**. Here is the prototype for strcpy: **char \*strcpy(char \*dest, const char \*src)**

**Question 2: (1 point): [POINTERS and TRACING CODE]**

What is the output of the following program?

#include <stdio.h>

void f(int\*, int );

int main()

OUTPUT:

{

int i=5, j=10;

f(&i, j);

printf("%d\n", i+j);

return 0;

}

void f(int\* p, int m)

{

m = m + 5;

\*p = \*p + m;

return;

}

**Question 3: (3 points) [FUNCTION POINTERS]**

Given the following function prototypes, complete the code listed in main.

void printNum(int);

int adder(int, int);

int main(int argc, char\* argv[])

{

1. Declare a function pointer that can be pointed to *printNum*
2. Initialize the function pointer from 1 above to point to *printNum*
3. Call *printNum* using the function pointer passing the value of 54

return 0;

}

**Question 4: (3 points) [CONSTANT VARIABLES VS #DEFINE]**

In class we discussed one major difference between C++ constant variables and C-style #define. Briefly describe this major difference. No more than 2 sentences.

**Question 5: (1 point each for a total of 3) [EMUMERATORS]**

Indicate whether each of the following enumerated data type definition is valid or invalid.

\_\_\_\_\_\_\_ enum Holiday {Easter, Halloween, Thanksgiving, Christmas};

\_\_\_\_\_\_\_ enum Holiday{“EASTER”, “HALLOWEEN”, “THANKSGIVING”, “CHRISTMAS”};

\_\_\_\_\_\_\_ enum Holiday{EASTER, HALLOWEEN, THANKSGIVING, CHRISTMAS};

**Question 6: (3 points)[POINTERS-TO-POINTERS and TRACING CODE]**

What is the output of the following code.

#include <stdio.h>

Output:

int f(int , int \*, int \*\*);

int main()

{

int c, \*b, \*\*a;

c = 5;

b = &c;

a = &b;

printf("%d \n", f(c, b, a));

return 0;

}

int f(int x, int \*py, int \*\*ppz)

{

int y, z;

\*\*ppz += 1;

z = \*\*ppz;

\*py += 6;

y = \*py;

x += 1;

return x + y + z;

}

**Question 7: (12 points)[MULTIPLE C++ CONCEPTS]**

**To save time, the lines bolded you can skip re-writing since they will not change.**

Rewrite the following program in C++

#include<stdio.h>

#define NUM 10

**int main(int argc, char\* argv[])**

**{**

FILE\* input = fopen(argv[1], "r");

FILE\* output = fopen("output.txt", "w");

**int array[NUM];**

**int sum = 0, i;**

**for(i = 0; i < NUM; i++)**

**{**

fscanf(input, "%d", &array[i]);

**sum += array[i];**

**}**

float average = ((float)sum)/NUM;

printf("average = %.2f\n", average);

fclose(input);

fclose(output);

**return 0;**

**}**

**Question 8: (1 point each for a total of 3)[C++ VARIABLES]**

Indicate which of the following variable declarations is valid. If the declaration is valid what is the declared data type?

Valid/Invalid variable declaration type

\_\_\_\_\_\_\_\_\_ auto var1 = 100; \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_ auto var2 = 456LL; \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_ auto var3; \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Question 9: (3 points each for a total of 6) [ENUMERATION]**

The following enumerators will cause a problem in your code.

**enum Color {RED, GREEN, BLUE};**

**enum Feelings {EXCITED, BLUE, MOODY};**

Describe the problem.

We discussed a new concept provided by C++ that will alleviate this problem. Rewrite the enums in a way that will alleviate the problem.

**Question 9: (4 points) [GENERAL C/C++ DATA READING ISSUES]**

Consider the program below:

#include <iostream>

**Explain the problem with this program. 2 points**

**Add the code to the program to fix this problem. Your solution must use the method we discussed in class. One line of code is all that is required here.**

**2 Points**

using namespace std;

int main(){

char ch;

cout << "Type a character and press Enter: ";

cin >> ch;

cout << "You entered " << ch << endl;

cout << "This program has paused. Press Enter to continue.";

cin.get(ch);

cout << "It has paused a second time. Please press Enter again.";

ch = cin.get();

cout << "It has paused a third time. Please press Enter again.";

cin.get();

cout << "Thank you!\n";

return 0;

}

When I compile and run this program I expect the program to do the following:

Type a character and press Enter: Y

You entered Y

This program has paused. Press Enter to continue.

It has paused a second time. Please press Enter again.

It has paused a third time. Please press Enter again.

Thank you!

However, this program actually ran as follows:

Type a character and press Enter: Y

You entered Y

This program has paused. Press Enter to continue. It has paused a second time. Please press Enter again.

It has paused a third time. Please press Enter again.

Thank you!

**Question 10: (2 points each for a total of 4) [SCOPE AND NAMESPACE]**

Consider the following namespace declarations. Below complete the two instructions above main.

#include <iostream>

using namespace std;

namespace foo

{

int fun() { return 5; }

}

namespace bar

{

const double PI = 3.1416;

double fun() { return 2\*PI; }

}

**1.In main write the code necessary to print the return value of the function fun() that belongs to foo.**

**2.In main write the code necessary to print the value of PI.**

int main ()

{

return 0;

}