**CPSC 1020 Fall 2018**

**EXAM #1**

**This is an individual exam. You may not use, books, phones, or any other outside resources. This exam consist of a total of 45 points, however, your grade will be graded on % basis. As an example, if you get 38 points correct your grade will be 38/45 which means you earned a grade of 84.44%**

**Question 1 (3 Points) [Function Pointers]**

Consider the following program:

#include <stdio.h>

void add(int a, int b){

printf("Addition is %d\n", a+b);

}

void subtract(int a, int b){

printf("Subtraction is %d\n", a-b);

}

void multiply(int a, int b){

printf("Multiplication is %d\n", a\*b);

}

int main(){

/\*In this program we need to declare an array of function pointers called **fun\_ptr\_arr.**  These function pointers should be pointed to the add, subtract and multiply functions shown above.  This can be done in one line of code.

Write the line of code that will declare the array of function pointers, as described above. Use an initialization list to point the pointers to the functions add, subtract and multiply. **THIS IS ONLY ONE LINE OF CODE. \*/**

int a = 15, b = 10;

/\*This is calling a function using an array of functions\*/

(\*fun\_ptr\_arr[2])(a, b);

return 0;

}

**Question 2 (1 Points) [String literals and arrays]**

Consider the following code:

#include <stdio.h>

int main()

{

char \*text\_pointer = "CPSC1020 Fall18!";

printf(" %c",text\_pointer[9]);

return 0;

}

What is the output?

1. This will only print a space
2. 8
3. CPSC1020
4. F
5. None of the above

**Question 3 (6 points Total) [C++ File Pointers]**

In class we discussed two ways to open a file pointer in C++.

Part 1 (2 point)

Write the code to open an input file pointer in C++ called **input1** using the 2 line approach. The file name will be supplied by argv[1].

Part 2(2 point)

Write the code to open an output file pointer in C++ called **output** using the 1 line approach. The file name will be supplied by argv[2].

Part3(2 point)

Write the code to open an input file pointer in C++ called **input2** using either approach. The file name will “test.txt”.

**Question 4 (3 points) [Pointers]**

Consider the following program:

#include <stdio.h>

int main(  ){

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

   char word[25] = {'H', 'a', 'v', 'e', ' ', 'a', ' ','g', 'r', 'e','a', 't', ' ', 'w', 'e', 'e', 'k', 'e', 'n', 'd', '!', '!', '!', '!'};

   int i, n=10;

What is the output of this program? You must write the entire output not just a number or character.

   int \*pv0 = values;

   printf(“pv0 = %i\n", \*pv0);

   int \*pv1 = pv0 + 3;

   printf("\npv1 = %i\n", \*pv1);

   char \*pv2;

   pv2 = word;

   printf("\n\*pv2 = %c\n", \*pv2);

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

   char \*pv4 = word;

If the output for this part includes a space us an underscore to indicate a space.

   for (i=0; i<3; i++) {

      printf("%c", \*pv4);

      pv4 += 3;

   }

   printf("\n%s\n", word);

   return(0);

}

**Question 5 (3 Points) [Pointers]**

In class, I discussed several reasons you would want to use pointers. **In no more than 2 sentences each, describe 2 of these reasons we specifically discussed in class.**

**Question 6: ( 9 points Total) [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 7 (2 Points Total)[Pointers]**

Multiple Choice. Consider the following declarations and initializations for the following questions.

int x = 3;

int y = 6;

int \*ptr1 = &x;

int \*ptr2 = &y;

1. What is the output of the following program fragment? (2 points)

printf(“%d %d\n”, x, \*ptr1);

printf(“%d %d\n”, y, \*ptr2);

1. 3 3

6 6

1. 3 &x

6 &y

1. 6 6

3 3

1. Based on the **original** initializations, what is the output of the following program fragment?

\*ptr1 = \*ptr2;

printf(“%d %d\n”, x, \*ptr1);

printf(“%d %d\n”, y, \*ptr2);

1. 6 6

3 3

1. 3 6

6 6

1. 6 6

6 6

1. 3 3

3 3

**Question 8 (6 Points) [Dynamically allocating memory]**

Consider the following c-style structs that represents the header and pixels of a PPM image:

struct header {

char MAGIC\_NUMBER[3];

int HEIGHT, WIDTH, MAX\_COLOR;

};

struct pixel{

unsigned char red;

unsigned char green;

unsigned char blue;

};

Assume you have read in the header information from the PPM image file. We now want to dynamically allocate the memory for the pixels, so we decided to create a function to do this for us. Write the code necessary to dynamically allocate the memory for a **2D** array of type **pixel**. You are to use the format we discussed at length in class. You should then return the array.

pixel \*\* allocateMemory(header hdr)

{

}

**Question 9 (2 point) [Double Pointers]**

Consider the following program and determine the output. It may be helpful to draw out the memory and pointers.

#include <stdio.h>

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

int main()

{

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

c = 7;

b = &c;

a = &b;

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

return 0;

}

Output:

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

{

int y, z;

\*\*ppz += 5;

z = \*\*ppz;

\*py += 3;

y = \*py;

x += 1;

return x + y + z;

}

**Question 10 (1 point) [Structures C-style vs C++ style]**

We discussed one major difference between C-style and C++ style structures. In no more than 2 sentences, explain the difference.

**Question** **11 (9 points) [C++ general]**

Consider the following C program. On the following page rewrite this program in C++.

#include <stdio.h>

#include <stdlib.h>

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

{

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

if(argc != 2)

{

printf("Execution error: <executable> <inputFile>\n");

exit(EXIT\_FAILURE);

}

if(input == NULL)

{

printf("The input file did not open properly\n");

return -1;

}

char c = getc(input);

while(c != EOF)

{

printf("%c", c);

c = getc(input);

}

fclose(input);

return 0;

}