**CPSC 1020 Summer 2017**

**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 54 points, however, your grade will be graded on % basis. As an example, if you get 44 points correct your grade will be 44/54 which means you earned a grde of 81.48%**

**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 (3 Points) C-Style printing**

Given the following program:

int main()

{

int a = 112;

int b = -1;

float c = 3.14;

int \*d = &a;

float \*e = &c;

**Write the c-style print statement that will print the address of a, b, c, and the values of d, and e.**

return 0;

}

**Question 3 (1 Points) String literals**

Consider the following code:

#include <stdio.h>

int main()

{

int i;

char \*text\_pointer = "Hello World!";

printf(" %c",text\_pointer[6]);

return 0;

}

What is the output?

1. This will only print a space
2. o
3. Hello
4. W
5. None of the above

**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?

   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;

   for (i=0; i<5; i++) {

      printf("%c", \*pv4);

      pv4 += 3;

   }

   printf("\n%s\n", word);

   return(0);

}

**Question 5 (3 Points) Constant Variables and C Style #define**

Both a C++ Constant Variable and a C Style #define are read only. They both must also be given a value when declared. In class we discussed a major difference between a C++ Constant Variable and a C Style #define. **In no more than 2 sentences explain the difference we discussed.**

**Question 6 (2 Points) Basics of classes**

Suppose your program contains the following class definition:

class Automobile

{

   private:

      double price;

      double profit;

   public:

      void setPrice(double newPrice);

      void setProfit(double newProfit);

      double getPrice();

};

Suppose main (driver) contains the following variable. ( You can assume these instances of the automobile class have been given default values. )

**Automobile hyundai, jaguar;**

Which of the following statements are acceptable when used in the main?

1. hyundai.price = 4999.99;
2. jaguar.setPrice(60500.00);
3. double aPrice = jaguar.getPrice();
4. if (hyundai.price == jaguar.price)

   cout << "Want to swap cars?";

**Question 7 (1 Points) Classes**

Suppose your program contains the following class definition:

class YourClass

{

   private:

      int information;

      char moreInformation:

   public:

//constructors

      YourClass(int newInfo, char moreNewInfo);

      YourClass();

//function

      void doStuff();

};

Based on the above, if in main (driver), which of the following would be a correct way to create an instance of the class called YourClass? (There could be more than one correct answer.)

1. YourClass anObject(42, ‘A’);
2. YourClass anotherObject;
3. YourClass yetAnothrObj(‘A’, 45);
4. YourClass andAnother(49.0, ‘B’);
5. Anobject = YourClass(99, ‘B’);

**Question 8 (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 9 (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 10 (3 Points) C-Style Stucture (struct)**

Consider the following C-style structure (struct)

struct Pixel

{

unsigned char red;

unsigned char green;

unsigned char blue;

};

The following is an example of how to create a variable of type Pixel.

struct Pixel pix;

Rewrite the C-style struct in a way that will allow me to create a variable of type Pixel in either of the following two ways:

struct Pixel pix;

Or

pixel\_t pix;

**Question 11 (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. (The same way you were required to use in PA1.) You should then return the array.

pixel \*\* allocateMemory(header hdr)

{

}

**Question 12 (1 points) OOP**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is a general model of something. It includes general characteristics of something without specific details.

**Question 13 ( 4 Points) Reference variables**

In class we discussed three differences between C++ References and C-style pointers.  One or more of the following are not true.  Below determine if each of the following are true or false.

Place T for true or F for false beside each of the following statements.

1. Both references and pointers must be initialized to Null if not connected to a piece of memory.
2. Once a reference is bound to legitimate variable it cannot be changed to reference a different variable.
3. Suppose I have the following:

int a = 10; int b =5;

int& aRef = a;

The following is OK to do:

aRef = b;

1. Suppose I have the following:

int a = 10; int b =5;

int& aRef = a; aRef = b;

aRef is now referencing b;

**Question 14 (1 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 = 8;

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 += 3;

Output:

z = \*\*ppz;

\*py += 4;

y = \*py;

x += 2;

return x + y + z;

}

**Question 15 (9 Points)**

Below is a simple “C” program. Using the following page rewrite this program in C++.

#include <stdio.h>

#include <stdlib.h>

#define SIZE 5

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

{

int numerator = 25;

int denominator = 10;

int sz;

int i = 0;

/\*

\*You can assume the files opened correctly and the correct number of

\*of command-line arguments were entered.

\*/

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

FILE \* outPut = fopen(argv[2], "w");

Result is 2.50

1.9871

39.9087

9.7658

51.3479

2.9813

float result = (float)numerator/denominator;

fprintf(outPut,"Result is %.2f\n", result);

float arr[SIZE];

for( ; i < SIZE; i++)

{

fscanf(inPut, "%f", &arr[i]);

fprintf(outPut, "%7.4f\n", arr[i]);

}

return 0;

}

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

{

Result is 2.50

1.9871

39.9087

9.7658

51.3479

2.9813

int numerator = 25;

int denominator = 10;

int i = 0;

float result =

float arr[SIZE];

for( ; i < SIZE; i++)

{

}

return 0;

}

**Question 16 (9 points) Classes**

Attached is the Person.h file that contains the declaration of the Person Class. Using this page and the following blank page, you are to implement this class. You should assume you are implementing the class in Person.cpp and not Person.h.

#ifndef PERSON\_H

#define PERSON\_H

#include <string>

#include <iostream>

using namespace std;

class Person

{

private:

string first;

string last;

int age;

public:

Person();

Person(string, string, int); //first, last, age

//setters

/\*This setter sets all information for the person\*/

void setAge(int);

void setFirst(int);

void setLast(int);

string getFirst();

string getLast();

int getAge();

/\*This function prints out the person’s name and age in the format of

\*Name: Yvon Feaster

\*Age: 54

This function is different than the one you did in lab. You are not returning a string so you will use cout not stringstream.

\*/

void printInfo();

};

#endif;