**CPSC 1020 Summer 2017 EXAM #1**

**You may not use, books, phones, online resources, or any other outside resources for this exam.  Please do not look at the computer of the person beside you or in front of you. There is a total of 110 points on this Exam (10 Extra credit points)**

**Question 1 (5 Points)**

Given the following snippet of code, answer the following question.

unsigned int a = 0b00111100;

unsigned int b = 0b00001101;

unsigned int c = 0;

c = a | b & 7;

What is the decimal (base 10) value of c?

**Question 2 (6 points)**

#include <stdio.h>

int main( ){

      char \*text\_ptr = "Good Morning!";

      printf("%c",text\_ptr[3]);

      text\_ptr[0] = 'A';

      for( ; \*text\_ptr ! = '\0'; ++text\_ptr)

           printf("%c", \*text\_ptr);

   return 0;

}

This code will compile.  However, when it is ran it will give a bus error, why?

**Question 3 (6 Points)**

Given the following:

#include <stdio.h>

/\*functions all with the same signature\*/

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. \*/**

unsigned int ch, a = 15, b = 10;

printf("Enter Choice: 0 for add, 1 for subtract and 2 for multiply\n");

scanf("%d", &ch);

if (ch > 2) return 0;

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

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

return 0;

}

**Question 4 (3 Points)**

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, d, and e.**

return 0;

}

**Question 5 (2 Points)**

Consider the following code:

#include <stdio.h>

int main()

{

int i;

char \*text\_pointer = "Good morning!";

printf(" %c",text\_pointer[3]);

What is the output?

1. O
2. d
3. Goo
4. None of the above

**Question 6 (6 points)**

Consider the following program:

#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));

   printf("\n4. Check: ");

   char \*pv4 = word;

   for (i=0; i<5; i++) {

      printf("%c", \*pv4);

      pv4 += 3;

   }

   printf("\n\n5. %s\n", word);

   printf("\nAll done!\n");

   return(0);

}

What is the output of this program?

**Question 7 (2 Points)**

Declare a “C-style” file pointer and open it for reading. The file name was passed in as the following command line arguments. (you are opening the input.txt)

./a.out output.txt input.txt

**Question 8 (2 Points)**

Declare a “C++ style” file pointer and open it for reading. The file name you are opening is input.txt.

**Question 9 (2 points)**

Supposed we opened a “C-style” file pointer called ***filePtr.***

Write the necessary code to close this file pointer.

**Question 10 (2 Points)**

Suppose we opened a “C++ style” file reference called ***filePtr.***

Write the code necessary to close this file pointer.

**Question 11 (5 Points)**

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 of #define. **In no more than 2 sentences discuss this difference.**

**Question 12 (8 Points)**

Suppose your program contains the following class definition:

class Automobile

{

   private:

      double price;

      double profit;

      double getProfit();

   public:

      void setPrice(double newPrice);

      void setProfit(double newProfit);

      double getPrice();

};

Suppose the main (driver) contains the following declaration and that the program somehow sets the values of all the member variables to some values:

**Automobile hyundai, jaguar;**

Which of the following statements are legal if used in the main?

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

   cout << "Want to swap cars?";

**Question 13 (6 Points)**

Suppose your program contains the following class definition:

class YourClass

{

   private:

      int information;

      char moreInformation:

   public:

      YourClass(int newInfo, char moreNewInfo);

      YourClass();

      void doStuff();

};

Based on the above, if in main (driver), which of the following would be a correct ways 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 14 (6 Points)**

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

**Question 15 (6 Points Total)**

Multiple Choice. Given 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? (2 points)

ptr2 = ptr1;

printf(“%d %d\n”, x, \*ptr1);

printf(“%d %d\n”, y, \*ptr2);

1. 6 6

6 6

1. 3 3

3 3

1. 3 3

6 3

1. 3 6

3 3

1. Based on the **original** initializations, what is the output of the following program fragment? (2 points)

\*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 16 (4 Points)**

In class we discussed two ways to create an inline function? In no more than 2 sentences each, describe the two ways to create an inline function.

A.

B.

**Question 17 (6 Points)**

Assume you will **not** know the number of **rows and columns** of a 2D array until run-time, therefore, we must dynamically allocate the memory for the 2D array.  Using the variables rows and columns, which both will be of type int, write the code to dynamically allocate the memory for a 2D array.

**Question 18 (3 Points)**

There are several errors in the following class declaration.

1.  class DumbBell;

2.  {

3.       private:

4.          int weight;

5.          DumbBell( );

6.          DumbBell(int);

7.       public:

8.          void setWeight(int);

9.  }

What lines have errors.

**Question 19 (2 points)**

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

**Question 20 (4 Points)**

In class we discussed three differences between C++ References and C-style pointers.  One or more of the following are not true.  Below mark the choices that are not true about references or pointers.

1. Both references and pointers must be initialized to Null if not connected to a piece of memory.
2. Once a reference is initialized to legitimate piece of memory it cannot be changed to reference a different piece of memory.
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 21 (2 Points)**

True/False

In this class we will focus on two common programming methods:

Procedural programming  (C)

Object Oriented (C++)

**Question 22 (10 points)**

This is a 2-part question.

Part 1:

Attached is the Person Class. You are to implement this class.

Part 2:

After you implement this class create a driver (main) that does the following.

1. Create an instance of the person class called myInfo. The constructor should assign your name and age to the appropriate class data members.
2. Print the information for the instance of myInfo.
3. Create a second instance of the class called myInstructor. Because I don’t want to give you my age, use the overloaded constructor leaving off my age.
4. Print the information for the instance of myInstructor.

NOTE: BE SURE TO READ THE INSTRUCTION WHEN WRITING THE PRINTINFO FUNCTION.

**Question 23 (10 Points)**

Attached you will find a simple “C” program. Rewrite this program in C++. This is the exact program you rewrote for the quiz.

#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

/\*Just in case someone does not want to reveal their age :-)\*/

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

//setters

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

void setPerson(string, string, int);

void setAge(int);

void setFirst(int);

void setLast(int);

string getFirst();

string getLast();

int getAge();

/\*This function prints out the persons name and age in the format of

\*Yvon Feaster is 53 If the person did not give their age then only print

\*their name.\*/

void printInfo();

};

#endif;

#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");

float result = (float)numerator/denominator;

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

float arr[SIZE];

for( ; i < SIZE; i++)

Result is 2.50

1.9871

39.9087

9.7658

51.3479

2.9813

{

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;

}