CPSC 2310 HOMEWORK

FUNCTION POINTERS/CALLBACK FUNCTIONS

Due: February 10, 2021, 11:59 Midnight

This homework assignment is a short assignment and should not take that long to complete. You will be required to watch two videos with a combined time of less than 30 minutes. You will then answer questions about the video and write a couple lines of code.

There are two parts to this Homework.

Part 1: 50% of the overall grade for this assignment
Watch the following videos and answer the following questions.
Function pointers:
https://www.youtube.com/watch?v=ynYtgGUNelE

What value does a function pointer hold?

The address of functions; they store the starting address or entry point of the block of memory containing all the instructions in a function.

Instructions of a program must be encode in binary.

The instructor in this video says that program instructions are executed in sequential order unless the instruction encounters, what?

When the instruction itself will say jump to another function.

Consider the following simple program and answer the questions within the code.

CPSC 2310 HOMEWORK

FUNCTION POINTERS/CALLBACK FUNCTIONS

Due: February 10, 2021, 11:59 Midnight

```
void printNum(int) {assume this function is implemented here};
void numSq(int) {assume this function is implemented here};
int adder(int, int) {assume this function is implemented here};

int main(int argc, char* argv[])
{
    /*Declare a function pointer called FP and point it to numSq*/
    void (*FP)(int) = &numSq;

    /*There are two ways to call a function pointer.  Show the two ways to call FP declared
above passing in the value of 10.  */
                int callPointer1, callPointer2;
                callPointer1 = (*FP)(10);
                callPointer2 = FP(10);

   /*The author of the video discussed two ways to point a function pointer to a function.*/
   /*Write the code to show the two ways to point FP to printNum (this should be 2 lines of
code.)*/
                *FP = &printNum;
                FP = printNum;


   /*Now that FP is pointing to a function, again there are two ways to call the function
Pointer/Below write the code to call the function pointer FP passing in the value of 20 using the
two methods.*/

                (*FP)(20);
                FP = 20;

  /*Declare a function pointer called FP2.*/

                int(*FP2)(int, int);

  /*Now point FP2 to adder;*/

                *FP2 = &adder;
```

```
 /*Now create a variable called sum and point it to FP2 passing in two integers. */
                void (*sum)();
                sum = &FP2(10, 15);


 printf("Sum = %d\n", sum);

 return 0;
}
```

CPSC 2310 HOMEWORK

FUNCTION POINTERS/CALLBACK FUNCTIONS

Due: February 10, 2021, 11:59 Midnight

Part 2: 50% of the overall grade for this assignment

Callback:

Watch the video below:

https://www.youtube.com/watch?v=sxTFSDAZM8s

This problem will use the concept of a callback function

Fill in the missing code:

```
#include <stdio.h>
#include <stdlib.h>

void fpTest(char*, char*);
void fpTest2(char*, char*);
void fpTest3(char*, char*);
void printStr(char*, char*, void(*fp)(char*, char*));

int main()
{
   /*Initialize these with your first and last names*/
   char* first = Annie;
   char* last =Hayes;

   /*Create a function pointer called fp and point it to fpTest*/
                void (*fp)(char*,char*);
                fp = &fpTest;

   /*Call printStr using first, last, and the function pointer you created
    *above*/
                 printStr(first, last, fp);


   /*Now point your function pointer to fptest2 */
                fp = &fpTest2;

   /*Call printStr using first, last, and the function pointer*/
                printStr(first, last, fp);

   /*Point your function pointer to fptest3*/
```

```
            fp = &fpTest3;

   /*Call printStr using two strings literals that you want to concatenate and the
    *function pointer. (not first and last)*/
            printStr("not first","not last", fp);
   return 0;
};

void fpTest(char* first, char* last)
{
  printf("First: %s\nLast: %s\n", first, last );
}

void fpTest2(char* first, char* last)
{
   printf("%s, %s\n", last, first);
}

void fpTest3(char* first, char* second)
{
   printf("%s%s\n", first, second);
}

void printStr(char* a, char* b, void(*fp)(char*, char*))
{
   fp(a, b);
}
```

Please make sure you understand these Two examples. Most likely, you have questions about this on one of your exams.

This is not required, but I encourage you to type these two programs in .c files and run them to test your answers. Most likely, you will use this concept in one of your programming assignments.