Sam Graham

P2 – COP 1220

9/11/2018

**1. What are the basic modules of a C program called?**

A program is a collection of one or more functions, one of which must be called main( ). A function has a header and a body. The header includes the function name and info about the type of information passed to and returned from the function. Functions have parentheses, which may be empty. The body of the function is enclosed in curly brace { } and consist of a series of statements, each terminated with a semi-colon (;). A declaration statement announces the name and type of variables used. Assignment statements give variables a value. Function call statements get the function to perform its programmed tasks. A return statement gives the calling function a return value.

**2. What is a syntax error? (Example 1 in English and 1 in C)**

A syntax error breaks the rules for C. It’s a grammatical error for the C language, just like a grammar error in English. Compilers are not as understanding as people. A person may be able to understand an English essay with grammatical errors. A compiler cannot interpret a C program with syntax errors. C syntax errors may use valid symbols in the wrong places.

English syntax error:

“Bugs (frustrating) can’ - become?”

Valid English symbols and words in improper places, but a person can likely understand what we are trying to say.

C syntax errors:

#include <stdio.h>

Int main(void)

[

/\* This program is full of syntax errors

Int a, b, c;

num1 = 6

num2 = 5 + a

num3 = a + b

Print(“If num1 = %d and num3 equals %d by adding num1 and num2 together, what is num1 –

num2?”, num1, num3;

]

The header capitalizes int when it shouldn’t. The code used brackets [ ] instead of curly braces { } to enclose statements within the function. The variable declaration uses capitalization for int again. The start comment block symbol /\* is used, but the end comment block \*/ is not used. Everything after the /\* is seen as a comment statement. Print function is mistyped, it should be printf(“…”)

**3. What is a semantic error? Example 1 in English and 1 in C**

Semantic errors are logic errors. You’ve got the syntax correct but the program is not performing as you intended. You’ve told it to do something you didn’t mean. Logic errors occur in English also. Calling out the wrong name in a crowd is likely to get a response from an unintended person.

**English semantic error:**

“Goodbye, I look forward to not swimming in the pool we just jumped in all day.”

Pretty sure goodbye is not appropriate, and if you just jumped in pool I hope you plan on swimming.

**C semantic error:**

#include <stdio.h>

int main(void)

{

// This program is supposed to divide two integers.

int num1, num2, num3;

num1 = 3;

num2 = 6;

num3 = num2 \* num1; // Here is the problem

printf(“By dividing %d by %d we get %d”), num2, num1, num3;

return 0;

}

Uh-oh, somebody multiplied instead of divided. The program does what we told it to, but not as we intended.

**4. C program that outputs names of favorite band with output included:**

// This is a simple program to display some

// of my favorite bands.

/\*

Written by Sam Graham on 9/8/2018 for COP 1220

I'm just learning so it's pretty basic.

\*/

/\*

I kept getting errors when writing to notepad and compiling with MinGW, so

I ended up using Visual Studio 2017 Community. Without "pch.h" included it would

not compile. Another auto-include for VS 2017 was <iostream>, but it's not needed for this program.

\*/

#include "pch.h"

#include <stdio.h>

void favoriteInstrument(void);

int main(void)

{

printf("My favorite band is Paul Thorn, he plays the ");

favoriteInstrument(); //Just practicing with functions and comments

printf("I also like CCR, The Beatles, Lynard Skynard, and that flute playing Jethro Tull.\n");

return 0;

}

void favoriteInstrument(void)

{

printf("guitar!\n");

}

**Output:**

My favorite band is Paul Thorn, he plays the guitar!

I also like CCR, The Beatles, Lynard Skynard, and that flute playing Jethro Tull.

**4. C program that computes and output the current age of the US in days from year 1776.**

/\*

This program displays the date and time in the current time zone, then asks the user to

input the current date. It runs through some loops counting up to

the current day from Jan 1, 1776. I'm still learning and couldn't figure out how to

auto calculate using a time\_t type.

Written by Samuel Graham. Took a few days of attempts, got it to work on 9/11/2018.

Used notepad and MinGW.

\*/

#include <stdio.h>

#include <time.h> // a quick search let me find where I can get current date time

#include <stdlib.h> // required for the exit functions. still learning more about exit functions.

int main()

{

time\_t current\_time;

char\* c\_time;

current\_time = time(NULL); // get the current time

if (current\_time == ((time\_t)-1)) //error checking if unable to get time.

{

(void)fprintf(stderr, "Failure to obtain the current time.\n");

exit(EXIT\_FAILURE);

}

c\_time = ctime(&current\_time); //set to the current time zone and made it a character string

if (c\_time == NULL) //more error checking, I have more to learn about these

{

(void)fprintf(stderr, "Failure to convert time to match your time zone.\n");

exit(EXIT\_FAILURE);

}

printf("Current date and time is %s\n", c\_time);

printf("This may help for your needed input\n");

printf(" \n"); // added space for readability

int days\_in\_month[] = { 0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 }; //array for day in month

int day, month, year, day\_counter;

int input\_day, input\_month, input\_year;

printf("Please input the 4 digit year: \a"); // alert escape isn't beeping like I had hoped.

scanf("%d", &input\_year); // obtains the year from the user, no data verification, so I hope they are nice.

printf("Please input the 2 digit month: \a");

scanf("%d", &input\_month); // obtain month from user, no data verification, please be friendly

printf("Please input the 2 digit day: \a");

scanf("%d", &input\_day); // obtain day from user, no data verification, please please don't hurt my program

day = 0;

month = 0;

year = 1776;

day\_counter = 0; //counts each day from Jan. 1, 1776 to the day input by user

while(input\_year > year)

{

day++; day\_counter++;

if (day > days\_in\_month[month]) // month variable designates which spot in the array we are counting

{ // there are 13 containers in this array. [0] has a value of 0 to accomodate.

day = 1; month += 1;

if (month > 12)

{

month = 1; year += 1;

}

}

}

while(month != input\_month)

{

day++; day\_counter++;

if (day > days\_in\_month[month])

{

day = 1; month += 1;

}

}

while(day != input\_day)

{

day++; day\_counter++;

}

printf("Our country is %d days old", day\_counter); // displays the message and the day\_counter

return 0;

}

**Output (Will change depending on what the user inputs):**

Current date and time is Tue Sep 11 11:34:18 2018

This may help for your needed input

Please input the 4 digit year: 2018

Please input the 2 digit month: 09

Please input the 2 digit day: 11

Our country is 88584 days old