EECS168:Lab8

Contents

Navigation

Home

Information

Syllabus
Schedule
Video Lectures
Exam Reviews

Classwork

Labs
Submitting Work

Due Time

This lab is due one week from your lab's start time.

Topics

- Functions

Functions

Functions are used to "bottle up," so to speak, some piece of functionality or a specific algorithm so that we can run it whenever we like and with whatever parameters we choose.

There are two parts to getting a function up and running. Step 1 is to define the function:

```
//template
<return type> <name> ( <parameters> )
{
   <function body>
   <return statement for return types other than void>
}
```

Example of a void function that prints "Hello" to the screen:

```
void printHello()
{
   std::cout << "Hello\n" << '\n';
}
```

Example of a function that takes a parameter and returns a value:

```
int succ( int n )
{
    int ans = n + 1; //optional step of declaring local variable
    return( ans );
}
```

You can do everything you already know how to do in your functions. Loops, if statements, complicated math - whatever you want.

If you want to use a function you defined, you to call it. You can call a function from main or another function.

Example:

```
int succ( int n )
{
    int ans = n + 1; //optional step of declaring local variable
    return( ans );
}

void printHello()
{
   std::cout << "Hello\n" << '\n';
}

int main()
{
   int x = 0;
   x = succ( 41 );//x is now 42

   std::cout << succ(54) << '\n'; //prints 55

   printHello();

   return(0);
}
```

void vs non-void

To decide if you need a void or non-void (e.g. a function that returns a value) you must determine if your function is returning a value or not. Decide whether these functions should have a return type or simply be void:

1. A function to find the cube of any int
2. A function that prints "Hello" 1000 times
3. A function that takes a string and a number and prints it took the screen that number of times
4. A function that calculates the nth fibonacci number
5. A function that takes a string called *phrase* and a file name as parameters. It writes that phrase to the file

Requirements

In this lab you'll make a single large program that uses several function definitions. Your main should have very little going on. In fact, I've provided your entire main below:

```
```

```
int main()
{
  run();
  return(0);
}
```

What your program will do is present the user with the following menu:

```
1) Count digits
2) Sum digits
3) Is Palindrome
4) Reverse
5) Exit
Choice:
```

Option	Behavior
Count digits	Prompt the user for a positive int and count how many digits are in that number
Sum digits	Prompt the user for a positive int and sum up all the digits in that number
Is Palindrome	Prompt the user for a positive int tell whether or not their number is a palindrome
Reverse	Prompt the user for a positive int and print the reverse of the number
Exit	Exit the program

For all choices, you may assume good input.

Required Functions

You will write several functions that you will use in combination to complete this lab:

- int lastDigit(int n)
  - returns the last digit in n
  - example lastDigit(123) returns 3
- int removeLastDigit(int n)
  - returns the number n with the last digit trimmed off
  - example removeLastDigit(123) returns 12
  - when removeLastDigit is passed a 1-digit number, return 0
- int addDigit(int currentNum, int newDigit)
  - returns the currentNum with newDigit placed into the ones space
  - example addDigit(123, 4) returns 1234
- int reverse(int n)
  - returns the reverse of n
  - example reverse(1234) returns 4321
  - this function should call other functions you've defined
- bool isPalindrome(int n)
  - returns true if n is a palindrome
  - example isPalindrome(12321) returns true, but isPalindrome(123) returns false
  - This should use reverse
- int countDigits(int n)
  - returns the count of digits in n
  - example countDigits(123) returns 3

    - this function should use removeLastDigit
- int sumDigits(int n)
    - returns the sum of digits in n
    - example countDigits(123) returns 6
    - this function should use removeLastDigit
- void printMenu()
    - prints the menu
    - This does NOT obtain user input, it only prints the menu
- void run()
    - Until the user wants to quit, this function will print the menu, obtain the user's choice, and call the appropriate functions to print the desired results

169: Is Prime

Add one more option for the user:

```
1) Count digits
2) Sum digits
3) Is Palindrome
4) Reverse
5) Is Prime
6) Exit
Choice:
```

If the user chooses this option, they will input a positive integer and your must tell them whether or not their number is prime. 1 is not considered prime for this program.

Rubric

- [5pts] Email subject line and file names correct

168

- [5pts] int lastDigit(int n)
- [5pts] int removeLastDigit(int n)
- [10pts] int addDigit(int currentNum, int newDigit)
- [10pts] int reverse(int n)
- [15pts] bool isPalindrome(int n)
- [15pts] int countDigits(int n)
- [15pts] int sumDigits(int n)
- [5pts] void printMenu()
- [15pts] void run()

169

- [5pts] int lastDigit(int n)
- [5pts] int removeLastDigit(int n)
- [10pts] int addDigit(int currentNum, int newDigit)
- [10pts] int reverse(int n)
- [15pts] bool isPalindrome(int n)

- [10pts] int countDigits(int n)
- [15pts] int sumDigits(int n)
- [5pts] void printMenu()
- [5pts] void run()
- [15pts] bool isPrime()

DO NOT put ANY other files in your package.
Designed by: John Gibbons