CSE100 – Assignment: Functions

Maximum Points: 50 points

Topics

- C++ API
- Functions – Value returning and void
- To declare and use variables
- Arithmetic Expressions
- Using cin to get the input from the user
- If-Statements
- Loops

Use the following Guidelines:

- Give identifiers semantic meaning and make them easy to read (examples numStudents, grossPay, etc).
- Keep identifiers to a reasonably short length.
- User upper case for constants. Use title case (first letter is upper case) for classes. Use lower case with uppercase word separators for all other identifiers (variables, methods, objects).
- Use tabs or spaces to indent code within blocks (code surrounded by braces). This includes classes, methods, and code associated with ifs, switches and loops. Be consistent with the number of spaces or tabs that you use to indent.
- Use white space to make your program more readable.

Important Note:

All submitted assignments must begin with the descriptive comment block. To avoid losing trivial points, make sure this comment header is included in every assignment you submit, and that it is updated accordingly from assignment to assignment. ***Note: There are 2 parts described – make sure you read all the pages***

Part 1: Written Exercise: (10 pts)

A) Explain the following function – what does it do? (3pts)

```
bool conf(char item)
{
      int val = (int)item;
      if((val >= 65) && (val <= 90))
            return true;
      else if((val >= 97) && (val <= 122))
            return true;
      else
            return false;
}
```

B) Explain the following function – what does it do? (3pts)

```
int place(char item)
{
     char temp = toupper(item);
     int index = (int)temp - 65;
     return index;
}
```

C) The following code is a function to calculate Fibonacci numbers recursively (4pts)

```
#include <iostream.h>
int Fibonacci(int x)
{
  if (x == 0) return 0;  // Stopping conditions
  if (x == 1) return 1;
  return Fibonacci(x - 1) + Fibonacci(x - 2);
}

int main()
{
  int num;
  cin >> num;
  cout << Fibonacci(num) << endl;
  return 0;
}
```

Trace through the recursive calls and show the result of calling `Fibonacci(5)`

Show each layer of calls and returns as its own line

```
Example:
Fibonacci(2)
Fibonacci(1) + Fibonacci(0)
1 + 0
1
```

Part 2: Programming (40 pts)

Write a C++ program called **LastName_FirstName_Assignment3.cpp (***eg: Doe_Jane_Assignment3.cpp***) --** The program is to display questions and read user inputs, then calculate and print out the requested value with a proper format. Don't forget your header block with your information and your comment block with the answers for part 1.

Description:

Create and test a series of simple functions.

Notes:

- Parameters and Arguments are the same thing
- Remember Void vs. Value returning functions

Instructions:

For this assignment you will create and test functions in C++. The functions that you must create and test are described in detail below.

List of Functions to Create:

1. A heading for the program including your name and your major at ASU. The output should look like this:
    - Name: Jo Doe
    - Major: Computer Science
    - Function Testing Program
2. A function that will prompt the user to input their name and return the string
3. A function that will take a name string as a parameter/argument and output a welcome message.
4. A function that will make use of other functions to figure out what day of the week it is:
    - A function that will display a menu of options to the user listing a number and the corresponding day of the week.
        - "1 - Monday" etc.
    - A function that will prompt the user for their menu selection and return the integer
    - A function that will take an integer and report the appropriate day
        - Pass in 1 - Prints out "So today is Monday!"
    - *Note: You should use loops to validate the input - if the user inputs >7 or <0 you should redisplay and re-prompt the user*
5. A function that takes an integer argument called count and a string argument called message. The function will display the message <count> times. For example, if the message is "Hello!" and the count is 3, then the functions will display "Hello!" three times.
6. A function that takes two int arguments called num1 and num2. The function will return the largest of the two argument values. If the arguments are equal, then it will return the first argument value.
7. A function that takes two Cartesian points of information (4 integers) and returns a double that is the distance between the two points.

You should create a routine to run all of these function in the above order. You should prompt the user when necessary and pass the correct information to the functions and use value returning functions as well. DO NOT USE GLOBALS

You may also create and use additional functions to help with the responsibilities of the ten required functions if you wish.

Sample output

```
Name: Jo Doe
Major: Computer Science
Function Testing Program
Enter your name here: Justin Selgrad
Welcome Justin Selgrad!

1 - Monday
2 - Tuesday
3 - Wednesday
4 - Thursday
5 - Friday
6 - Saturday
7 - Sunday
Please enter the number of a day (1-7): 5
So today is Friday!

Please enter a string: This message will repeat three times
Please enter how many times it should repeat: 3
This message will repeat three times
This message will repeat three times
This message will repeat three times

Please enter two integers: 8 9
Of 9 and 8, 9 is larger.

Please enter x1: 5
Please enter y1: 5
Please enter x2: 10
Please enter y2: 10
The distance is 7.0710
```