 

 

 



**Programming Challenges (Ch#5: if-then-else statements & Ch#6: Introducing Loops) – WEEKS 7-8 in-class/lab**

//Program Shell1 prompts for and reads a one-digit number.

// Values between 0 and the digit (inclusive) are summed.

#include <iostream>

using namespace std;

int main ()

{

int counter; // loop-control variable

int sum; // running sum

int digit;

cout << "Enter a one-digit number; press return."

<< endl;

cin >> digit;

counter = /\* TO BE FILLED IN \*/

sum = /\* TO BE FILLED IN \*/

while ( ) /\* TO BE FILLED IN \*/

{

/\* TO BE FILLED IN \*/

}

cout << "Sum of digits between 0 and "

<< digit << " is " << sum << endl;

return 0;

}

1. **Sum of Numbers (SAMPLE 1)**

Write a program that asks the user for a positive integer value. The program should use a loop to get the sum of all the integers from 1 up to the number entered. For example, if the user enters 5, the loop will find the sum of 1,2,3,…….5.  
 What type of a loop is this?

Input Validation: Do not accept a negative starting number.

1. **Sum of Numbers (SAMPLE 2a)**

Write a program that asks the user for several integer values and ends when the user inputs a -1. The program should use a loop to sum of all the positive integers entered. For example, if the user enters 5, 3, 2, -5, 6, 8,-1 the loop will find the sum of 5, 3, 2, 6 & 8.

What type of a loop is this?

1. **Sum of Numbers (SAMPLE 2b)**

Write a program that asks the user for a data file with several integer values and ends when a negative value (for example -5) is found. The program should use a loop to sum of all the positive integers in the file. For example, if the file contains 5, 3, 2, -5, 6, 8,-1 the loop will find the sum of 5, 3, 2.

What type of a loop is this?

Input Validation: Include necessary data validation routines to check for state of the file stream.

Prior to working on this challenge it is recommended that you review the class hand out: Input\_Processing\_FileIO.docx

**More programming challenges to practice writing various types of loops (REPETITION CONTROL STRUCTURE) WK7**

1. **The Least of These**

Write a program with a loop that lets the user enter a series of integers, followed by -9 to signal the end of the series. After all the numbers have been entered, the program should display the smallest number entered.

1. **The Greatest of These**

Write a program with a loop that lets the user enter a series of integers, followed by -9 to signal the end of the series. After all the numbers have been entered, the program should display the largest number entered.

1. **The Greatest and Least of These**

Write a program with a loop that lets the user enter a series of integers, followed by -9 to signal the end of the series. After all the numbers have been entered, the program should display the largest and the smallest numbers entered.

1. **Find Duplicates**

Write a program with a loop that lets the user enter a series of integers, followed by -9 to signal the end of the series. After all the numbers have been entered, the program should display the following information:

1. Total count of whole numbers entered
2. The total number of 9’s entered or the lack thereof
3. The first and last position where a 9 was entered.

-------------------------------------------------------------------------------------------------------------------------------------------------------  
**Algorithm Workbench Exercise#4 – – LOGIC ROUTINES USING NESTED LOOPS WITH IF-ELSE STATEMENTS**

**EXTRA CREDIT (10pts) - DUE MONDAY MARCH 7, 2023**

As we continue with our study of programming fundamentals, here is a short extra credit programming challenge involving repetition control structures. To be specific, the program specifications below and the algorithm you develop and write will involve the set-up and use of both while and do-while loops.

**Random Number Guessing Game**

Write a program that generates a random number between 5 and 15 and asks the user to guess what the number is. If the user’s guess is higher than the random number, the program should display ***Too high. Try again***. If the user’s guess is lower than the random number, the program should display ***Too low, Try again***. The program should use a loop that repeats while keeping a count of the number of guesses the user makes until the user correctly guesses the random number. Then the program should display the number of guesses along with the following message ***Congratulations. You figured out my number***. Suggest that you also give the user the opportunity to play the game again or quit.

(HINT: See [career\_predictor2.cpp](http://www.sonic.net/sarkar/cs10sum2017/wks6_8/career_predictor2.cpp) for a sample set-up and implementation of random numbers that maybe helpful)

Thanks, Sujan!

Sujan Sarkar, Instructor Santa Rosa Junior College Computer Studies Department Maggini Hall, 524-1873

Email: [ssarkar@santarosa.edu](mailto:ssarkar@santarosa.edu)

Note: Since this short programming exercise is "optional" i.e.., not a required assignment and if you have completed assignments 1- 6, the 10 extra credit points earned here will be added as bonus points to your midterm examination point tally for the Spring 2023 session.

/\*career\_predictor.cpp - Given three career options...this program predicts a most likely career

using random numbers and an if/else if statement.\*/

#include <iostream>

#include <string> // Needed to use strings

**#include <cstdlib> // Needed for random numbers**

using namespace std;

int main()

{

// Strings to hold user entered careers

string career1, career2, career3;

**int randomNum; // Will hold the randomly generated integer**

**// "Seed" the random generator**

**unsigned seed = time(0);**

**srand(seed);**

// Explain the game and get the user’s career choices

cout << "Good to see you are running my free career predictor program.\n"

<< "Please enter 3 careers you would like to have. Example: \n\n"

<< " chef \n astronaut \n CIA agent \n\n"

<< "Then I will predict what you will be. \n\n";

cout << "Career choice 1: ";

getline(cin, career1);

cout << "Career choice 2: ";

getline(cin, career2);

cout << "Career choice 3: ";

getline(cin, career3);

**// Randomly generate an integer between 1 and 4.**

**randomNum = 1 + rand() % 4;**

// Use branching logic to output the prediction

if (randomNum == 1)

cout << "\nYou will be a " << career1 << ". \n";

else if (randomNum == 2)

cout << "\nYou will be a " << career2 << ". \n";

else if (randomNum == 3)

cout << "\nYou will be a " << career3 << ". \n";

else

cout << "\nSorry. You will not be any of these. \n";

return 0;

}