CS 110: Lab 04

Repetition Statements-Loops

Muhammad Mujtaba

CMD ID: 540040

mmujtaba.bese25seecs@seecs.edu.pk

Class: BESE 16B

Batch: 2k25

Task 1 [CLO 2]: Number Guessing Game

CODE

```cpp
#include <iostream>
#include <random>

// get random value of type T (can be any type) between given values inclusive
template <typename T>
T getRandom(T minInc, T maxInc)
{
    std::random_device rd;  // get a random device
    std::mt19937 gen(rd()); // using a random engine
    // "uniform_int_distribution" used to remove random bias
    std::uniform_int_distribution<> distrib(minInc, maxInc);
    // return the number
    return distrib(gen);
}

int main()
{
    // variables
    size_t guess_counter = 0; // size_t is commonly used for counters in cpp
    unsigned int secret = getRandom<int>(1, 10); // not negative; <int> is not
required but used to emphasize that template function is used
    unsigned int guess; // not negative

    // MAIN LOOP
    std::cout << "I have chosen a number between 1 and 10. Try to guess it.\n\n";
    do // do while is used to intialize the guess of user
    {
        std::cout << "Your guess: ";
        std::cin >> guess;
        // newline by using pressing enter

        // increase stored guess counter
        guess_counter++;

        // if guessed correctly then print and exit
        if (secret == guess)
        {
            std::cout << "\n\nThat's right! You guessed it. \n";
            std::cout << "It only took you " << guess_counter << " tries.\n";
            break;
        }

        // if not executed -> guess is wrong
        std::cout << "\nThat is incorrect! Guess Again.\n";
    } while (secret != guess);

    // ignoring previous input
    std::cin.ignore();
    std::cin.get();
    return 0;
}
```

OUTPUT

```
I have chosen a number between 1 and 10. Try to guess it.

Your guess: 4

That is incorrect! Guess Again.
Your guess: 7

That is incorrect! Guess Again.
Your guess: 2

That is incorrect! Guess Again.
Your guess: 9


That's right! You guessed it.
It only took you 4 tries.
```

Task 2 [CLO 2]: Computing Harmonic Mean

CODE

```cpp
#include <iostream>

int main()
{
    int input_n = 0;
    // using double for more accuracy
    double result_numerator = 0.f;
    double result_denominator = 0.f;

    std::cout << "Enter the amount of inputs: ";
    std::cin >> input_n;
    // newline by user pressing enter

    result_numerator = input_n;

    // take `input_n` values from user
    int counter = 1;
    do
    {
        float x_i;
        std::cout << "Enter x" << counter << ": ";
        std::cin >> x_i;
        // newline by user pressing enter

        result_denominator += 1.f / x_i;
        counter++;
    } while (counter <= input_n);

    std::cout << "Harmonic mean if your input numbers is " << result_numerator /
result_denominator << "\n";

    // ignoring previous input
    std::cin.ignore();
    std::cin.get();
    return 0;
}
```

OUTPUT

```
obscure@Obscures-MacBook-Air output % ./"task2"
Enter the amount of inputs: 3
Enter x1: 1
Enter x2: 2
Enter x3: 3
Harmonic mean if your input numbers is 1.63636

obscure@Obscures-MacBook-Air output % ./"task2"
Enter the amount of inputs: 10
Enter x1: 1
Enter x2: 2
Enter x3: 3
Enter x4: 4
Enter x5: 5
Enter x6: 6
Enter x7: 7
Enter x8: 8
Enter x9: 9
Enter x10: 10
Harmonic mean if your input numbers is 3.41417

obscure@Obscures-MacBook-Air output % ./"task2"
Enter the amount of inputs: 4
Enter x1: 1
Enter x2: 2
Enter x3: 3
Enter x4: 4
Harmonic mean if your input numbers is 1.92
```

Task 3 [CLO 3] Factorial:

CODE

```cpp
#include <iostream>

int main()
{
    int input_n = 0;
    int factorial = 1; // intializing to 1 instead of 0

    std::cout << "Input a number to find factorial of: ";
    std::cin >> input_n;
    // newline by user pressing enter

    // if user user inputs 0 program doesn't run and returns stored value of
`factorial` i.e 1
    // start from 1 and multulpy all numbers till `input_n`; store result in
`factorial`
    for (int i = 1; i <= input_n; i++)
    {
        factorial *= i;
    }

    std::cout << "Factorial of " << input_n << " is " << factorial << "\n";

    // ignoring previous input
    std::cin.ignore();
    std::cin.get();
    return 0;
}
```

OUTPUT

```
obscure@Obscures-MacBook-Air output % ./"task3"
Input a number to find factorial of: 5
Factorial of 5 is 120

obscure@Obscures-MacBook-Air output % ./"task3"
Input a number to find factorial of: 10
Factorial of 10 is 3628800

obscure@Obscures-MacBook-Air output %
```

Task 4 [CLO 3] Find lowest priced item:

CODE

```cpp
#include <iostream>
#include <string>
#include <limits>
#include <iomanip>

int main()
{
    int num_items = 0;
    std::string lowest_name = "";
    // float is not a simple type like int so we cannot just flip all bits to one, that is NaN
    float lowest_price = std::numeric_limits<float>::max(); // storing max value that can be
stored in float

    std::cout << "Number of items to Enter: ";
    std::cin >> num_items;
    std::cout << "\n";
    std::cout << std::setw(32) << std::setfill('-') << "\n";
    std::cout << "\n";

    for (int i = 1; i <= num_items; i++)
    {
        std::string name = "";
        float price = 0;

        std::cout << i << ". Enter the Item Name: ";
        std::cin.ignore(); // ignoring previous input
        std::getline(std::cin, name);
        // getline reads until newline from the given stream (std::cin in this case) and
stores it in the given strings

        std::cout << i << ". Enter the Item Price: ";
        std::cin >> price;
        // newline by user pressing enter

        // compare to already stored lowest value, if this is lower then update lowest
        // no need to compare names
        if (price <= lowest_price)
        {
            lowest_name = name;
            lowest_price = price;
        }

        // spacer for next item
        std::cout << std::endl;
    }
    std::cout << "\n";
    std::cout << std::setw(32) << std::setfill('-') << "\n";
    std::cout << "\n";
    std::cout << "Name of lowest Item: " << lowest_name << "\n";
    std::cout << "Price of lowest Item: " << lowest_price << "\n";

    // ignoring previous input
    std::cin.ignore();
    std::cin.get();
    return 0;
}
```

OUTPUT


obscure@Obscures-MacBook-Air output % ./"task4"
Number of items to Enter: 3
1. Enter the Item Name: Apples
1. Enter the Item Price: 20
2. Enter the Item Name: Oranges
2. Enter the Item Price: 10
3. Enter the Item Name: Grapes
3. Enter the Item Price: 30
Name of lowest Item: Oranges
Price of lowest Item: 10
obscure@Obscures-MacBook-Air output %