p4.18

Henry Hsu

October 5, 2011

1 Specification

Prime numbers. Write a program that prompts the user for an integer and then prints out all prime numbers up to that integer. For example, when the user enters 20, the program should print

2 Analysis/Design

Utilizing brute force approach, test each possible positive integer up to user selected number to see if it is a prime number.

Resource prime number.

```
for {$m = 2 \to \sqrt n$}
   if $n \mod m$
      prime_status = false$
   else
      Prime_status = true$
$i \gets i + 1$
```

3 Implementation

"p4_18.cpp" 1≡

⟨ *Include files* 2c ⟩

```
int main()
{
 ⟨ input number 2a ⟩
 ⟨ generate prime numbers 2b ⟩
}
```

◇

⟨ *input number* 2a ⟩ ≡

```
int target_number;
cout << "Please enter a positive integer: ";
cin >> target_number;                    // store user input into variable
if (cin.fail())
{
        cout << "Your input was not an integer." << endl;
        return 1;
}

if (target_number < 0)
{
        cout << "Please input only positive integers." << endl;
        return 1;
}
```

◇

Fragment referenced in 1.

⟨ *generate prime numbers* 2b ⟩ ≡

```
bool prime_status;
for (int prime_numbers = 2; prime_numbers <= target_number; prime_numbers++)
        {
        prime_status = true;
        for(int i = 2; i < prime_numbers; i++)
                {
                        if (prime_numbers % i == 0) prime_status = false;
                }
        if (prime_status) cout << prime_numbers << endl;
}
```

◇

Fragment referenced in 1.

These are the include files needed for library function calls

⟨ *Include files* 2c ⟩ ≡

```
#include <iostream>
using namespace std;
```

◇

Fragment referenced in 1.

4 Test

Using a test value of 20 should yield: 2, 3, 5, 7, 11, 13, 17 and 19 as results.