1DV533 STEP 1 Assignment report

Jesper Malmberg em222vs@student.lnu.se

Task 3

I simply used a while loop and modulo operator to check which integers where odd and then multiplied the total with this.

```
//--------------------------------------------------------------------------
//--------------------------------------------------------------------------
// File: Step_3.cpp
// Summary: This program calculates the product of all odd integers 1-15
// Version: 1.1
// Owner: Jesper Malmberg
//--------------------------------------------------------------------------
// Log: 2021-11-26 Created file
//--------------------------------------------------------------------------
// Preprocessor directives
#include <iostream>
#include <iomanip>
using namespace std;
// Prototypes
void intProduct();
int main() {

    intProduct();
    return 0;
}
//---------------------------------------------------------------------
// void intProduct()
// Calculates the product of all odd integers 1-15
//---------------------------------------------------------------------
void intProduct() {
    int x = 0;
    int product = 1;
    // Iterate 15 times
    while (x < 16) {
        // Check if number is odd
        if (x % 2 != 0) {
            product = product * x;
        }
        x++; // Increase x by 1 for next iteration
    }
    cout << "The product of all odd integers 1-15 is: " << product;
}
```

Task 4

I used a foor loop to iterate 1 -15 and and printed 1 multiplied by 1, 10, 100 and 1000 in columns using setw() command.

```
//-------------------------------------------------------------------------
//-------------------------------------------------------------------------
// File: Task_4.cpp
// Summary: This program prints multiplication tables
// Version: 1.1
// Owner: Jesper Malmberg
//-------------------------------------------------------------------------
// Log: 2021-11-26 Created file
//-------------------------------------------------------------------------
// Preprocessor directives
#include <iostream>
#include <iomanip>
using namespace std;
// Prototypes
void getTable();
int main() {

    getTable();
    return 0;
}
//-------------------------------------------------------------------
// void getTable()
// Prints a table with 1-15 multiplied by 1, 10, 100 and 1000
//-------------------------------------------------------------------
    void getTable() {
    cout << "Multiplication table" << endl;
    cout << "====================" << endl << endl;
    // Setup the basic table printout
    cout << setw(5) << "n" << setw(10) << "10*n" << setw(10) << "100*n" <<
setw(10) << "1000*n" << endl
    << "---------------------------------" << endl;
    // The table
    for (int x = 1; x <= 15; ++x) {
        cout << setw(5) << x*1 << setw(10) << x * 10 << setw(10) << x * 100 <<
setw(10) << x * 1000 << endl;
    }
}
```

Task 5

I used nested loops to iterate “9 rows and 9 columns”. Again I used modulo operator to get the staggered pattern.

```
//-------------------------------------------------------------------------
//-------------------------------------------------------------------------
// File: Task_5.cpp
// Summary: This program prints a pattern of * symbols
// Version: 1.1
// Owner: Jesper Malmberg
//-------------------------------------------------------------------------
// Log: 2021-11-27 Created file
//-------------------------------------------------------------------------
// Preprocessor directives
#include <iostream>
#include <iomanip>
using namespace std;
// Prototypes
void printPattern();
int main() {

    printPattern();
    return 0;
}
//-------------------------------------------------------------------
// void printPattern()
// Prints a pattern of * symbols
//-------------------------------------------------------------------
void printPattern() {
    // Outer for-loop controls the rows of the "table"
    // Two inner for-loops prints the "columns" based on if i is even or odd
    for (int i = 0; i < 9; i++) {
        if (i % 2 == 0) {
            // Check if i is even or odd
            for (int j = 0; j < 9; j++) {
                cout << "* " << " ";
            }
            cout << endl;
        }
        else {
            for (int k = 0; k < 9; k++) {
                cout << " *" << " ";
            }
            cout << endl;
        }
```

```
    }
}
```

Task 6

Basically I just used a while loop to enter 10 integers. Upon entry I checked if the entered on was higher than the highest in which case the stored highest becomes second highest and the entered number is stored as the new highest.

```
//---------------------------------------------------------------------------
//---------------------------------------------------------------------------
// File: Step_6.cpp
// Summary: This program compares 10 integers and prints the second highest
// Version: 1.1
// Owner: Jesper Malmberg
//---------------------------------------------------------------------------
// Log: 2021-11-27 Created file
//---------------------------------------------------------------------------
// Preprocessor directives
#include <iostream>
#include <iomanip>
using namespace std;
// Prototypes
void compareIntegers(int);
// Global Variables
int highest = 0;
int secondHighest = 0;

int main() {

    int counter = 0;    // Counter for the inner while-loop
    char answer;        // Answer whether or not to continue
    int input;          // The input as entered by the user
    do {
        cout << "Enter 10 different integers to check which is the second
highest." << endl;
        // Repeat 10 times
        while (counter < 10) {
            cout << "Enter an integer and press enter: ";
            cin >> input;
            compareIntegers(input);
            counter++;
        }
        cout << "The second highest entered integer is: " << secondHighest<<
endl;
```

```
        cout << "Do one more time (Y/N)?" ;
        cin >> answer;
        counter = 0;
        highest = 0;
        secondHighest = 0;
    } while (answer == 'Y' || answer == 'y');
    return 0;
}
//-----------------------------------------------------------------
// void compareIntegers()
// Compares entered integers to one another and stores highes and
// second highest
//-----------------------------------------------------------------
void compareIntegers(int input) {
    // If input is higher than stored highest integer the input becomes
    // the highest and the old highest becomes second highest
    if (input >= highest) {
        secondHighest = highest;
        highest = input;
    // If the input is between secondHighest and highest the input becomes
    // the second highest
    } else if (input > secondHighest) {
        secondHighest = input;
    }
}
```

Task 7

The key was to initialize the highest and lowest at its theoretical limit of 0 and 10.0 so that I later in the conditional statements can check where the entered number belongs. Once all numbers are entered I simply divide by 7 to get the average score.

```
//-------------------------------------------------------------------------
//-------------------------------------------------------------------------
// File: Step_7.cpp
// Summary: This program inputst 9 doubles, removes the highest and lowest
// then returns and prints the average of the remaining 7 numbers
// Version: 1.1
// Owner: Jesper Malmberg
//-------------------------------------------------------------------------
// Log: 2021-11-27 Created file
//-------------------------------------------------------------------------
// Preprocessor directives
#include <iostream>
#include <iomanip>
```

```
using namespace std;
// Prototypes
void checkInput(double);
// Global Variables
double highest = 0;     // Initialize at 0 in order to have correct reference
double lowest = 10.1;   // Initialize minimum above max theoretical score of 10.0
to have a reference value
double total = 0;       // The total score
double counter = 0;     // Counter for the inner while-loop and used for average
calculation

int main() {
    char answer;        // Answer whether or not to continue
    double input;       // The input as entered by the user
    do {
        cout << "Enter 10 different integers to check which is the second
highest." << endl;
        // Repeat 9 times
        while (counter < 9) {
            cout << "Enter the score and press enter: ";
            cin >> input;
            checkInput(input);
            counter++;
        }
        cout << "The average score is: " << total / (counter - 2)  << endl; //
Divide total score by 7
        cout << "Do one more time (Y/N)?";
        cin >> answer;
        counter = 0;
        highest = 0;
        lowest = 0;
        total = 0;
    } while (answer == 'Y' || answer == 'y');
    return 0;
}

//---------------------------------------------------------------
// void checkInput()
// Compares entered doubles to one another and stores highest and
// lowest. The values between gets added to total
//---------------------------------------------------------------
void checkInput(double input) {
    // If higher than highest (starting out as 0), input becomes highest and
whatever was highest gets added to the total
    if (input > highest) {
```

```
        total += highest;
        highest = input;
    }
    else if (input < lowest) {
        // If lowest has been "initialized" add the previous minimum to the total
score
        if (lowest < 10.1) {
            total += lowest;
        }
        // Input becomes the lowest value
        lowest = input;
    }
    // If neither highest or lowest just add to total score.
    else {
        total += input;
    }
}
```

Task 8

Nested for loops, the inner one checks if the active number from the outer one is divisible by another number, if it is it's not a prime and the loop breaks. If it's not possible to divide it is a prime number, and is printed to the console. A Boolean isPrime keeps track whether or not the active number is a prime.

```
//---------------------------------------------------------------------------
//---------------------------------------------------------------------------
// File: Task_8.cpp
// Summary: This program finds and prints all the prime numbers 2-100
// Version: 1.1
// Owner: Jesper Malmberg
//---------------------------------------------------------------------------
// Log: 2021-11-28 Created file
//---------------------------------------------------------------------------
// Preprocessor directives
#include <iostream>
#include <iomanip>
using namespace std;

int main() {
    // Outer loop goes through all numbers 2-100
    for (int i = 3; i <= 100; i++) {
        bool isPrime = true;    // Initialize to True until we can verify that i
is not a prime
        // Inner loop checks if i is a prime
        for (int j = 2; j < i; j++) {
```

```
            // If we get a match, ie i divides with no rest on j break the loop
            if (i % j == 0) {
                isPrime = false;
                break;
            }
        }
        // If i doesn't divide without rest on j we have a prime and print the
answer
        if (isPrime) {
            cout << i << endl;
        }
    }
}
```

Task 8

Unfortunately I ran out of time and couldn't complete this task.