C++ Week 1 Lecture 1

After Lecture 1 of Week 1 the student should know:

* The difference between managed and unmanaged programming languages
* What Just In Time compilation is
* When a C++ program is compiled
* Where the starting point of a C++ program is
* How to create a new empty C++ project
* How to create a basic program
* How to print text to the console
* How to create a variable
* That a variable should be initialized
* How to create a function
* Where to place a function
* How to call a function
* How to make sure a function can return a value
* How to give one or more variables to a function as input
* How to create an array
* How to fill an array
* How to reach a certain element in an array
* How to give an array to a function as input
* How to do basic arithmetic operations
* How to make a for-loop
* How to make a while-loop
* How to make a do-while-loop

Exercises for Lecture 1.1

For every exercise you are only allowed to include <iostream>. You can create all functions in the same file. Make sure that you test each function with the appropriate function-call from ‘main()’.

1. Create a function that gets two integers as input and returns the sum.
2. Create a function that gets an integer as input and prints on the console whether it is “even” or “uneven”.
3. Create a function that gets two integers ‘base’ and ‘exponent’ as input, and recursively(!) calculates ‘base’ to the power of ‘exponent’. It can be assumed that ‘exponent’ is not negative.
4. Create a function that gets one integer as input and returns whether or not the number is prime. If needed, you can include <cmath> to calculate the square root.
5. Create a function that gets an array of integers and a single integer named ‘value’ as input and returns the first index that contains ‘value’. If it is not in the array, the function returns ‘-1’.
6. Create a function that gets an array of integers and an integer ‘size’ as input and uses bubble-sort to sort the numbers in the array ascending. (For those who forgot bubble-sort, there is a link in the link-section)
7. Create a function that gets two integers as input and uses Euclid’s algorithm to calculate the greatest common divisor or GCD. The GCD of two numbers is the greatest number that is a divisor for both. (See the link-section for a link to the Wikipedia page of Euclid’s algorithm)
8. Create a function that gets two integers as input and uses the function for the greatest common divisor to calculate the least common multiple or LCM. The LCM of two numbers is the smallest number that is a multiple of both.   
   For each pair of integers (x, y) it is true that: GCD(x, y) \* LCM(x, y) = x \* y.

Useful Links for Lecture 1.1

Thorough tutorial about functions:  
<http://www.cplusplus.com/doc/tutorial/functions/>

Bubble sort:  
<https://interactivepython.org/runestone/static/pythonds/SortSearch/TheBubbleSort.html>

Euclid’s algorithm:  
<https://en.wikipedia.org/wiki/Euclidean_algorithm>