**Name**: Steve Hommy

**Pair: -**

**Amount of completed tasks: 10**

**Which tasks were left undone or incomplete: 0**

Self-assessment:

This exercise was easy for me because I have been coding with Python for many years and these tasks were beginner level. I didn’t really learn anything, but it was fun to recap these kinds of tasks. So in summary I understood everything.

## Test report

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Input / action** | **Desired output** | **Actual output (use red color if desired output != actual output)** |
| **1** | Run code | Hello | Hello |
|  |  |  |  |
| **2** | E.g. numbers: 2, -5, 4, 7, 9, 11, 0, 445, -100, 4  E.g. strings: abc, 34re5, word, qwerty, cat-doc, def, 4, #-!?bc, alkf, oooooo | Integers: 2, -5, 4, 7, 9, 11, 0, 445, -100, 4  Strings: : abc, 34re5, word, qwerty, cat-doc, def, 4, #-!?bc, alkf, oooooo | Enter number of elements in list: 10  Enter number: 2  Enter number: -5  Enter number: 4  Enter number: 7  Enter number: 9  Enter number: 11  Enter number: 0  Enter number: 445  Enter number: -100  Enter number: 4  Type anything: :  Type anything: abc  Type anything: 34re5  Type anything: word  Type anything: qwerty  Type anything: cat-doc  Type anything: def  Type anything: 4  Type anything: #-!?bc  Type anything: alkf  [2, -5, 4, 7, 9, 11, 0, 445, -100, 4]  [':', 'abc', '34re5', 'word', 'qwerty', 'cat-doc', 'def', '4', '#-!?bc', 'alkf']  [68, 97, 38, 61, 63, 49, 61, 38, 60, 28] |
|  |  |  |  |
| **3** | E.g. numbers: 2, -5, 4, 7, 9, 11, 0, 445, -100, 4 | Arranged list: -100, -5, 0, 2, 4, 4, 7, 9, 11, 445  (Don’t worry about the formatting, it may be without commas as well.) | Enter number of elements in list: 10  Enter number: 2  Enter number: -5  Enter number: 4  Enter number: 7  Enter number: 9  Enter number: 11  Enter number: 0  Enter number: 445  Enter number: -100  Enter number: 4  Type anything: abc  Type anything: 34re5  Type anything: word  Type anything: qwerty  Type anything: cat-doc  Type anything: def  Type anything: 4  Type anything: #-!?bc  Type anything: alkf  Type anything: ooooo  [-100, -5, 0, 2, 4, 4, 7, 9, 11, 445]  ['#-!?bc', '34re5', '4', 'abc', 'alkf', 'cat-doc', 'def', 'ooooo', 'qwerty', 'word'] |
|  |  |  |  |
| **4** | User inputs integers 5, -34 and 0. | Please give an integer: 5  Please give an integer: -34  Please give an integer: 0  Number of negative integers is: 1 | Please give an integer: 5  Please give an integer: -34  Please give an integer: 0  Number of negative integers: 1 |
|  |  |  |  |
| **5** | User inputs integers 16, -34, 17, 0. | Please give an integer: 16  Please give an integer: -34  Please give an integer: 17  Please give an integer: 0  Number of even integers is: 2 | Please give an integer: 16  Please give an integer: -34  Please give an integer: 17  Please give an integer: 0  Number of even integers: 2 |
|  |  |  |  |
| **6** | User inputs integers -3, 7, 30, 9, 0. | Please give an integer: -3  Please give an integer: 7  Please give an integer: 30  Please give an integer: 9  Please give an integer: 0  Sum of positive integers divisible by three is: 39 | Please give an integer: -3  Please give an integer: 7  Please give an integer: 30  Please give an integer: 9  Please give an integer: 0  Sum of positive integers divisible by three is: 39 |
|  |  |  |  |
| **7** | User inputs 13 | Give maximum value: 13  Procession is: 3, 6, 9, 12  Number of terms is: 4  Sumf of terms is: 30  Sum of squared terms is: 270 | Give maximum value: 13  Procession is: [3, 6, 9, 12]  Number of terms is: 4  Sum of term is: 30  Sum of squared terms is: 270 |
| **7** | User inputs 0 | Give maximum value: 0  Procession is:  Number of terms is: 0  Sumf of terms is: 0  Sum of squared terms is: 0 | Give maximum value: 0  Procession is: []  Number of terms is: 0  Sum of term is: 0  Sum of squared terms is: 0 |
| **7** | User inputs -15 | Give maximum value: -15  Procession is:  Number of terms is: 0  Sumf of terms is: 0  Sum of squared terms is: 0 | Give maximum value: -15  Procession is: []  Number of terms is: 0  Sum of term is: 0  Sum of squared terms is: 0 |
|  |  |  |  |
| **8** | User inputs R  User inputs P  User inputs S  User inputs S  User inputs P | Give your choice (R, P, S): R  Computer’s choice is Paper.  Paper covers Rock.  Computer 1 You 0  Give your choice (R, P, S): P  Computer’s choice is Paper.  It’s a tie!  Computer 1 You 0  Give your choice (R, P, S): S  Computer’s choice is Paper.  Scissors cuts Paper.  Computer 1 You 1  Give your choice (R, P, S): S  Computer’s choice is Rock.  Rock crushes Scissors  Computer 2 You 1  Give your choice (R, P, S): P  Computer’s choice is Scissors.  Scissors cuts Paper  Computer 3 You 1  You lost! | Give your choice (R, P, S): R  Computer's choice is S  Rock crushes Scissors  Computer 0 You 1  Give your choice (R, P, S): P  Computer's choice is S  Scissors cuts Paper  Computer 1 You 1  Give your choice (R, P, S): S  Computer's choice is P  Scissors cuts Paper  Computer 1 You 2  Give your choice (R, P, S): S  Computer's choice is R  Rock crushes Scissors  Computer 2 You 2  Give your choice (R, P, S): S  Computer's choice is P  Scissors cuts Paper  Computer 2 You 3  You win! |
|  |  |  |  |
| **9** | Run program multiple times | Random number is: 5  Random number is: 1  Random number is: 6 | Random number is: 4  Random number is: 2  Random number is: 5 |
|  |  |  |  |

**10**

Explain the following terms (use your own words, do not copy paste answers from Internet). You can answer in Finnish or English.

1. **Procedural programming**

Procedural programming is a programming paradigm built around the idea that programs are sequences of instructions to be executed. Programming paradigms are a way to classify programming languages based on their features. Languages can be classified into multiple paradigms. Procedural programming focus heavily on splitting up programs into named sets of instructions called procedures, analogous to functions.

1. **Functional programming**
2. **Object oriented programming**
3. **Class (in programming)**
4. **Object (in programming)**
5. **Instance (in programming)**
6. **Encapsulation (in programming)**