1.1P: Preparing for OOP – Answer Sheet

# Introduction

This paper ’s answer sheet serves two purposes:

1. It serves as a revision for you of your previous learnings; and
2. It establishes a baseline understanding of your knowledge in key Computer Science topics.

As such this paper is divided into the following areas of knowledge:

1. Your experience with UNIX/DOS console commands;
2. Your ability to differenKate between data types (e.g. text) and informaKon categories (e.g. Ktle);
3. Your experience with compiler parsing and evaluaKon of expressions according to rules of precedence (e.g. BODMAS, also known as GEMS or PEMDAS);
4. Your understanding of Computer Science concepts and various compiler constructs such as blocks and scope;
5. Finally taking three steps, we want you to develop a program as follows:
   1. starKng with a simple funcKon: you provide the pure logic and calculaKons, no input, nor output;
   2. Then, in the second step, you write the main line code that invokes that simple funcKon. Your main line code will provide the necessary data, and then you will print out the result of the funcKon’s calculaKon.
   3. Finally we want you to add business logic to the main line program’s code; that business logic will interpret the results of the funcKon, and inform your user with informaKon about the results.

## Section A: Console commands

1. Explain the following terminal instrucKons:

1. cd: change directory
2. pwd: print working directory show the directory are currently at
3. mkdir: make directory it will create new directory
4. cat: display file content
5. ls: show the content of the current directory

## Section B: Data types and Information categories

1. Consider the following categories of informaKon, and suggest the most appropriate data type to store and represent each kind of informaKon:

|  |  |
| --- | --- |
| **Informa$on Category** | **Suggested Data Type** |
| A person’s family name | string |
| A person’s age in years | interger |
| A person's weight in Kilograms | float |
| A telephone number | interger |
| A temperature on the Kelvin scale | float |
| The average age of a group of children | float |
| Whether the student passed this task | boolean |

1. Aside from the examples already provided above, please come up with your own examples of informaKon that could be stored as:

|  |  |
| --- | --- |
| **Data Type** | **Suggested Information Category** |
| String | Text message |
| Integer | Calories count |
| Float | height |
| Boolean | Check if something is correct |

**Sec$on C: Compiler evalua$on of expressions**

1. Fill out the **last** two columns of the following table based on the expression and values we have supplied.
2. Evaluate the value of each expression under column 1, given its formula, values, and variables; use the given values (column 2) of any variable(s) in the expression.
3. IdenKfy the value of the results (column 3), and the data type the result is most likely to be (column 4) in a complier “friendly” form (e.g. Float):

|  |  |  |  |
| --- | --- | --- | --- |
| **Expression** | **Given** | **Result** | **Data Type** |
| 76 |  | 76 | interger |
| True |  | True | boolean |
| a | a = 3.1415927 | 3.1415927 | float |
| 1 + 2 \* 3 + 4 |  | 11 | interger |
| a and False | a = True | False | boolean |
| a or False | a = True | True | Boolean |
| a + b | a = 1 b = 3 | 4 | Interger |
| 3 \* a | a = 5 | 15 | Interger |
| a \* 2 + b | 1. = 2.5 2. = 3 | 8 | Float |
| a + 2 \* b | 1. = 2.5 2. = 3 | 8.5 | Float |
| (a + b) \* c | a = 2  b = 4  c = 6 | 36 | Interger |
| “Fred” + “ Astair” |  | Fred Astair | String |
| a + “ Rogers” | a = “Ginger” | Ginger Rogers | String |

**Sec$on D: Compiler Constructs and CS Concepts:**

1. Using some code as an example, please explain the difference between **declaring** and **initializing** a variable.

The difference between the two is declaring only introduce a variable and specify it type while initializing also put a value into the variable

*Paste your example code below:*

Int declaring

Int initializing = 19

1. Explain the term **parameter**. Write some **code** that demonstrates a simple of use of a parameter. You should show a procedure or funcKon that uses a parameter, and how you would call that procedure or funcKon.

A parameter is a variable that is passed into a function,method or procedure to provide input. Help to reduce hardcode

*Paste your example code below:*

Def Draw(parameter):

Print(parameter);

Draw(“a cute little cat that’s is not an image”)

1. Using an **coding example**, describe the term **scope** as it is used in procedural programming (not in business nor project management). Make sure you explain the differences of as many kinds of scope that you can idenKfy (at least two, and up to five).

Scope is…*<finish the sentence(s)>*

*Paste your example code below:*

## Section E: Implementing Algorithms, Data Handling, and Informing Results - Personalized Requirements

**STEP 1:**

1. In a procedural style, in any language you prefer, write a funcKon called Average, which accepts an array of integers, and returns the average of those integers.
2. **Do not use any libraries for calcula$ng the average**: we want to see your understanding of algorithms.
3. You must demonstrate appropriate use of parameters, returning and assigning values, and the use of loop(s). **Note — just write the func$on at this point.** In the next step we will ask you to ***invoke the func-on.***
4. You should **not** have a complete program, **nor** even code that outputs anything at this stage. This is a **func$on**; and input/output and any business logic processing is the responsibility of the (main line) calling code.

*Paste your example func.on code below:*

**STEP 2:**

1. Using the same preferred language, write the main line calling code you wouldneed to (a) marshal the data, (b) invoke the funcKon, (c) print out the result, and (d) **print out your student name and student Id**
2. We do **not** require you to provide any input processing logic; you sim ply have provide the inline instanKate of a collecKon of data values (provided below) for the funcKon to calculate the average of that data set.

a. Sample data values

2.5, -1.4, -7.2, -11.7, -13.5, -13.5, -14.9, -15.2, -14.0, -9.7, -2.6, 2.1 7. Note: your should have made **no changes** to your funcKon.

*Paste all of your example code below:*

*Paste your example code’s output here:*

1. Using the same preferred language, add to your exisKng main line code above, the following business logic code for interpreKng the result of the funcKon’s calculaKons.
2. Print the message “MulKple digits” if the average is above or equal to 10. Otherwise, print the message “Single digits”.

10.And then, if the average is negaKve, add an addiKonal line of output staKng “Average value negaKve”.

11.Finally, if the last digit of the average is larger than the last digit of your Student ID, please print the message "**Larger than my last digit**". Otherwise, please print the correct message, either "**Equal to my last digit**" or "**Smaller than my last digit**".

12.Note, you should not have made any changes to your implemented function13.Provide evidence of your program running, i.e. the code, its environment, and its run Kme outputs.

*Paste your example code’s output here:*

*Finally on a new page paste a SINGLE screenshot of your program (main line and func.on) running with its outputs here:*

# End of Task

**Please render your paper as a PDF and submit via CANVAS.**