## **COMP 1516 – Programming Fundamentals with Python – Assignment 1**

|  |  |
| --- | --- |
| **Total Marks** | 30 |
| **Due Dates** | TBD |

**Overview**

For this assignment you will be generating one of three different reports for one of three different inputs.

Input Types:

* Phone Data
* Tablet Data
* Laptop Data

Report Types:

* Text
* CSV
* JSON

The input type and report type will be specified as command line parameters to the Python script (main.py) that you will be creating. You are provided with a Python module (data.py) that has functions that return a list of data for each of the above input types. You will also create a Python module (reports.py) with functions for generating each of the above report types.

Your assignment must have no syntax errors and must successfully run (i.e., no runtime errors) for all combinations of the above inputs and outputs. **If the script cannot be run, you will receive a mark of zero on this assignment.**

Full marks will only be given to implementations that are reasonably efficient, i.e., there should NOT be a lot of copy and paste code.

**Requirements**

The following table identifies the requirements for this assignment.

|  |
| --- |
| **Requirement** |
| The command line parameters are as follows:  **main.py <input type> <report type>**  Where:   * Input Type is phone, tablet or laptop * Report Type is text, csv or json   The following errors should be printed to the console for invalid parameters:   * Input type must be either phone, table or laptop * Report type must be either text, csv or json   *Make sure you follow best practices for having a main function in this script.* |
| Create a module called reports in a reports.py file. Creates functions for the following three reports in this module: text report, csv report and json report. |
| The output format for the text report should be as follows:  Timestamp: <date time or report>  Device: <device type – Mobile Phone, Tablet or Laptop>  Number: <number of unique devices in list>  Average Price: $<calculated average price>  Minimum Price: $<minimum price in list>  Maximum Price: $<maximum price in list>  Median RAM: <median ram> GB  Operating Systems: <comma separated list of operating systems>  Used formatted strings to generate the csv output.  *The operating system should be the operating system value AND the version (i.e., Android 5.1.1 Lollipop)*  *The datetime format should be YYYY-MM-DD HH:MM* |
| The output format for the csv report should be as follows:  <datetime>,<device type>,<number of devices>,<calculated average price>,<minimum price in list>,<maximum price in list>,<median ram>,<slash separated list of operating systems>  Use the join function to generate the csv output.  *The operating system should be the operating system value AND the version (i.e., Android 5.1.1 Lollipop)*  *The format of the list of unique operating systems should look like:*  *Android 5.1.1 Lollipop/Android 7.1 Nougat/Android 9.0 Pie*  *The datetime format should be YYYY-MM-DD HH:MM* |
| The output format for the json report should be as follows:  {  “data\_time”: “<date time of report>”,  “device\_type”: “<device type>”,  “number”: <number of devices>,  “average\_price”: <calculated average price>,  “min\_price”: <minimum price in list>,  “max\_price”: <maximum price in list>,  “median\_ram”: <median ram in gb>,  “operating\_systems”: [  “<os 1>”,  “<os 2>”,  …  ]  }  Optional: Use the json module to generate the json output.  *The operating system should be the operating system value AND the version (i.e., Android 5.1.1 Lollipop)*  *The datetime format should be YYYY-MM-DD HH:MM.* |
| Correct Output for each input type and report type (see *Example Run Configurations to Test* below).  Full marks will be given for correct output and efficient implementation (limited redundant code). |

**Example Run Configurations to Test**

There are nine run configurations that must be tested:

* main.py phone text
* main.py phone csv
* main.py phone json
* main.py tablet text
* main.py tablet csv
* main.py tablet json
* main.py laptop text
* main.py laptop csv
* main.py laptop json

**Sample Output**

The **sample\_output.txt** file provides the expected output for each of the above run configurations. Your script must produce the same output.

**Submission**

Submit the following Python files in a **zipfile** called **assignment1.zip** to the Assignment 1 dropbox in D2L:

* main.py – Your implementation of the main script
* reports.py – Your implementation of a reports module
* data.py – Provided with the lab

Make sure you followed naming best practices and included DocString in all functions.

**Grading Summary**

|  |  |
| --- | --- |
| Implementation  As per the requirements described above AND the test\_main.py unit test. | 30 marks |
| Marks will be subtracted for **each** violation of naming standards or missing DocString. |  |
| **Total** | **20 marks** |

**Best Practices**

* Variable names should be lower\_snake\_case
* Function names should be lower\_snake\_case
* All functions should include a DocString comment