Python 2020-2 HW1

Submit ‘hw1\_YourStudentID.py’ to HISNET > ASSINMENTS > HW1

**Unit Converter**

1. When the program starts, print what this program is for and how to use.
2. Show the list of services for the unit conversion. Ask the user to choose one of the services.
   1. The number of services must be at least 7. (ex. angle, currency, length, temperature…)
   2. Provide the service number as well as the service name. Users are required to type the service number to choose a specific unit conversion service.
   3. If the user's input is wrong (ex. empty, spaces, non-numeric type, ...) or out of the service range, print out an appropriate message to tell the user what's wrong. Then terminate the program.
   4. The termination procedure: (1) print the termination message at the last line. (2) Wait until the user confirm it (hit the enter key). (3) When the user hits the enter key, the program will be terminated.
3. If a service category is selected, show the sub-menu for the service. Ask the user to choose one of the menu items.
   1. The number of submenu items for each service category must be at least 3. (ex. If the service category is length, then the submenu can be ‘meters to millimeters’, 'kilometers to miles’, 'centimeters to inches', and etc.)
   2. Provide the submenu item number as well as the item name. Users are required to type the item number to execute a specific unit conversion.
   3. If the user's input is wrong (ex. empty, spaces, non-numeric type, ...) or out of the item range, print out an appropriate message to tell the user what's wrong. Then terminate the program. The termination procedure is same with 2-D.
4. Ask the user to enter the value to convert.
   1. Before asking the user to enter the value, print out the message to show the user the options that he/she chosen and to tell what he/she is supposed to enter as a value. Carefully compose the message to guide the user to enter an appropriate form of input.
   2. Only a positive integer is considered as valid. Otherwise, print out an appropriate message to tell the user what's wrong. Then terminate the program. The termination procedure is same with 2-D.
   3. If the input value is valid, convert the unit and show the result to the second decimal points with an appropriate message.
5. Variable names and code readability 1 point
6. Comments (at least 80% of the codes) 1 point