Assignment 04

Due: Monday, Oct 26th, 2020 at 10:30 a.m.

General Assignment Notes

Your assignments must follow the following requirements:

- Name your Eclipse project with:
  - your network login,
  - an underscore,
  - 'a' (for 'assignment'),
  - the assignment number: `login_a#`.

  For example, if student `barn4520` submits Assignment 1, the name should be: `barn4520_a1` or `barn4520_a01`. Make sure your programs and the testing file are in this Eclipse/PyDev project.

- In your program, use the variable naming style given in Coding and Documentation Style Standards : i.e. lower case variable names, underscores between words.

- Test your programs:
  - Copy the output from your test to a file in your Eclipse/PyDev project named `testing.txt`.
  - Make sure that you have included an identification header for each question; there was a sample identification header for the testing file in lab 1.
  - Make sure you do the correct number of tests as specified in the question.
  - Make sure the tests are well labeled so that the markers know how the results match the questions.
  - The solutions for all programs go into *one* `testing.txt` file.

- Zip the entire project using Eclipse.
  - Give your .zip file the same name as your project when exporting your project, e.g. `barn4520_a1.zip` .
  - Use only Eclipse's built-in archive capability to create these .zip files. No other format will be accepted.

- Use the Validate Assignment link to make sure that your project and .zip file are named correctly and have the proper contents. Improper assignment submissions are given a grade of zero.

- Submit the validated .zip file to the appropriate drop box on MyLearningSpace.
  - You can submit as many times as you like. Only the last submission is kept.
- Unless otherwise indicated by the question you may only use the built-in functions and special forms introduced in the lecture slides
- The solutions you submit must be entirely your own work. Do not look up either full or partial solutions on the Internet or in printed sources.

Assignment 04

Due: Monday, Oct 26th, 2020 at 10:30 a.m.

General Marking Expectations

- Although the marking scheme is tailored for each question, you can use the following as an indication of what we are looking for when marking your assignments.
    - General:
        - project and zip files named correctly. Using the wrong names is an automatic zero.
        - uses the variable naming style given in Coding and Documentation Style Standards : i.e. lower case variable names, underscores between words
    - main:
        - identification template included and filled in correctly
        - inputs as required for program
        - outputs as required for program
    - testing:
        - identification header included and filled in correctly
        - Tests as required for program, e.g. number of tests, types of tests.

Note: Refer back to Lab 1 for help in the process of submitting an assignment.

Assignment 4 marking Expectations:

1. All programs should use the formatted output method taught in lab 2.
2. Define constants when appropriate.
3. The outputs for all of the questions in an assignment go into one testing.txt
4. function(s):
    1. **FUNCTIONS MUST BE IN A SEPARATE FILE FROM THE MAIN PROGRAM.**
    2. for questions that specify the objectives of the function(s), did you do what was requested?
    3. correct parameters and return values
    4. calculation(s) correct
    5. function documentation has the correct format
    6. Parameters correct and complete
    7. Returns correct and complete
5. main:
    1. identification template included and filled in correctly
    2. proper use of constants if appropriate
    3. inputs as required for program
    4. outputs as required for program
    5. outputs should use the formatted output method
    6. functions calls have appropriate arguments
    7. return values from functions are appropriately handled
6. testing:
    1. identification header included and filled in correctly
    2. tests as required for program, e.g. number of tests, types of tests.
7. **Provide the docstring for all questions**

Assignment 04

Due: Monday, Oct 26th, 2020 at 10:30 a.m.

1. A manufacturing company measured the productivity of its workers and found that:
   - between the hours of 6am and 10am they could produce 30 pieces/hour/worker
   - between 10am and 2pm they could produce 40 pieces/hour/worker
   - between 2pm and 6pm they could produce 35 pieces/hour/worker.

   Write a function `pieces_produced` takes a number that represents a starting hour between 6am and 6pm, in twenty-four hour format (between 0 and 23), along with the number of workers and returns a number that represents the total number of pieces produced during that hour. If the user enters an invalid hour or number of workers, the function should return -1. Save the function in a PyDev library module named `functions.py`

   Write a testing program named **t01.py** that tests the function by asking the user to enter hours and number of workers and prints the total number of pieces produced.

   A sample run:

```
Time of the day: 6
Number of workers: 10
The total number of pieces produced:300

Or
Time of the day: 35
Number of workers: 10
Can not perform the calculation
```

   - test your program with 3 different data than the examples, you may assume that the user will only enter numbers.
   - Copy the results to testing.txt.

2. Write a function `num_day` that takes a number in the range of 1 through 7 as a parameter and returns a string representing the corresponding day of the week, where 1=Monday, 2 =Tuesday, 3=Wednesday, 4 = Thursday, 5 = Friday, 6 = Saturday, and 7 = Sunday. The function should return the string "`Error`" if the parameter is that is outside the range of 1 through 7. You may assume that the parameter will be only numbers. Save the function in a PyDev library module named `functions.py`

   Write a testing program named **t02.py** that tests the function by asking the user to enter a number and displaying the output.

   A sample run:

```
Please enter a number between 1 and 7: 7
The number 7 corresponds to Friday.
```

Or

```
Please enter a number between 1 and 7: 9
The number 9 corresponds to Error.
```

- Test your program with all possible input.
- Copy the results to `testing.txt`.

3. There are two types of taxes: Federal tax and province tax.
   1. Federal taxes:

      Assume that the share of income $35,000 and below is federally taxed at 15%, the share of income between $35,000 and $100,000 is taxed at 25%, and the share of income over $100,000 is taxed at 35%.

      (i.e an income over$ 100,000 is taxed as follows:

      - The portion of income up to and including the first $35,000 is federally taxed at 15%.

      - The portion of income between $35,001 and $100,000, inclusive, is federally taxed at 25%.

      - The income earned beyond $100,000 is federally taxed at 35%

   2. Province taxes: Assume that there is no provincial income tax for income up to $50,000 but is a flat 5% for all income above $50,000.

   Write 2 functions to calculate total tax liability for ***any*** income.
   - Function `fed_tax` that takes one parameter `income` as an input and return a number that represents the calculated federal tax.
   - function `prov_tax` that one parameter `income` as input and return a number that represents the calculated province tax.

   Write a testing program named **t03.py** that tests the functions by asking the user to enter an income and displays the calculated taxes. Save the function in a PyDev library module named `functions.py`

   A sample run:

```
Enter your income: $ 200000
Federal tax:    $56500.00
Provincial tax: $ 7500.00
Total tax:      $64000.00
```

   - Test your program with three different income data, you may assume that the user will only enter numbers
   - Copy the results to `testing.txt`.

Assignment 04

Due: Monday, Oct 26th, 2020 at 10:30 a.m.

4. On a roulette wheel, the pockets are numbered from 0 to 36. The colours of the pockets are as follows:
   - Pocket 0 is green.
   - For pockets 1 through 10, the odd-numbered pockets are red and the even-numbered pockets are black.
   - For pockets 11 through 18, the odd-numbered pockets are black and the even-numbered pockets are red.
   - For pockets 19 through 28, the odd-numbered pockets are red and the even-numbered pockets are black.
   - For pockets 29 through 36, the odd-numbered pockets are black and the even-numbered pockets are red.

   Write a function, `pocket_colour` that takes an integer as parameter and returns the colour corresponds to the pocket number. "The function should return the string "Error" if the user enters a number that is outside the range of 0 through 36. Save the function in a PyDev library module named `functions.py`

   Write a testing program named **t04.py** that tests your function by asking the user to enter a pocket number and displays whether the pocket is green, red, or black.

   A sample run:

```
Enter a pocket number: 0
The selected pocket is green.
```

   - Test your program with 10 different data, you may assume that the user will only enter numbers.
   - Copy the results to testing.txt.

5. The colours red, blue, and yellow are known as the primary colours because they cannot be made by mixing other colours. When you mix two primary colours, you get one of following secondary colour:
   - Mix red and blue: purple
   - Mix red and yellow: orange
   - Mix blue and yellow: green

   Write a function `colour_mix` that takes two strings as parameters as two primary colours and returns the secondary colour. If the parameters are anything other than "red", "blue", or "yellow", the function should return the string "Error". Save the function in a PyDev library module named `functions.py`

   Write a main program named **t05.py** that tests the function by asking the user to enter the names of two primary colours to mix and displays the mixed colour.

   A sample run:

```
Enter a first colour: red
```

Assignment 04

Due: Monday, Oct 26th, 2020 at 10:30 a.m.

```
Enter a second colour: blue
The mixed colour is purple.

Or

Enter a first colour: red
Enter a second colour: brown
The mixed colour is Error
```

- Test your program with 2 different data, other than the example.
- Copy the results to testing.txt.