

## CS21 Assignment #4

Please use the given name for your scripts.

1. (*pennies.py*) Complete Programming Exercise #7 in chapter4 (page 162). *Validate the input to ensure the number of years is 1 or greater.* When formatting your data columns, do not use the tab (\t) (it is ok to use spaces and tabs to format the column headings which are strings).

In addition to computing and printing the total pay, also calculate and display the average daily wage, as shown.

### Sample Run #1:

```
Number of days worked: 10
```

```
Salary Earned Each Day
```

| Day | Amount (\$) |
|-----|-------------|
| --- | -----       |
| 1   | 0.01        |
| 2   | 0.02        |
| 3   | 0.04        |
| 4   | 0.08        |
| 5   | 0.16        |
| 6   | 0.32        |
| 7   | 0.64        |
| 8   | 1.28        |
| 9   | 2.56        |
| 10  | 5.12        |

```
Your total pay = $10.23
```

```
Your average daily wage = $1.02
```

### Sample Run #2:

```
Number of days worked: 0
```

```
Enter number of days >= 1: -5
```

```
Enter number of days >= 1: 3
```

```
Salary Earned Each Day
```

| Day | Amount (\$) |
|-----|-------------|
| --- | -----       |
| 1   | 0.01        |
| 2   | 0.02        |
| 3   | 0.04        |

```
Your total pay = $0.07
```

```
Your average daily wage = $0.02
```

A few items to note:

- Look carefully at spacing of output. (no space between dollar sign and total pay/average page)
- You must use the format function for printing the data in the table. Tabs/spaces may be used to format column headings only.

2. (primes.py) A prime number is defined as an integer greater than 1 that has no positive divisors other than 1 and itself. Write a program that prompts the user for an integer > 1. Validate the value is > 1 (if not, ask for another). Use a loop to determine if the number is prime or not. Issue an appropriate message. [complete this part before proceeding].

Add a loop that continues to ask the user if they would like to enter more numbers to test. See sample runs.

#### Sample Run #1

Welcome to my prime number detector.  
Provide an integer and I will determine if it is prime.

```
Enter an integer > 1: 5
5 is prime.
Do you want to try another number? (Y/N) : n
```

#### Sample Run #2

Welcome to my prime number detector.  
Provide an integer and I will determine if it is prime.

```
Enter an integer > 1: 0
Input must be > 1, try again: 10
10 is not prime.
Do you want to try another number? (Y/N) : y
Enter an integer > 1: 1
Input must be > 1, try again: 0
Input must be > 1, try again: 13
13 is prime.
Do you want to try another number? (Y/N) : y
Enter an integer > 1: 5
5 is prime.
Do you want to try another number? (Y/N) : n
```

A few items to note:

- If the user enters anything but 'y' or 'Y' when asked if they want to do it again, the loop terminates.
- A list of prime numbers to test your program with: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53. (Please note that your program must work for other prime numbers as well)

#### Reminders (not following will result in point deductions):

Use constants! No magic numbers!

It is expected that you will complete the same process of development that use in class. When you reach the point of having an algorithm (pseudocode), this will become the comments of your program as a starting point for writing code. Comment first, then code!

Be sure to include comments at the top of the program that include your name, class and a short description of the program.

Be sure all output is formatted. Unless otherwise, specified, displays non-integer values with 2 digits after the decimal point.

*Any work you submit for this assignment should be authored entirely by yourself. Assistance is permitted from the instructor or teaching assistants only. All submitted programming assignments are subject to originality verification through software designed and used for the Measure Of Software Similarity (MOSS).*