

CIS 125 Principles of Programming Logic

Test #2: Decision Structures

100 points

Directions

Complete the following **three** Python programs.
Program #1 is worth 60 pts. Program #2 is worth 25 pts. Program #3 is worth 15 pts.

Late submissions cannot be accepted. The exam is **open book, open note, open Internet.** However, you **cannot receive live assistance from anyone** or post questions on discussion forums, i.e. any resources you must already exist.

Academic Honestly Policy

Students are expected to uphold the school's standard of conduct relating to academic honesty.

Program #1: Change Program: dollars and cents (60 pts)

- Create a Python program named `test2-1.py`
- This program will allow a user to enter **any dollar** and **cents** amount and then will **provide change**.
- Add a **Conditional statement** to make sure they did not enter **\$0.00**, if they do, then display a message saying, **"You must enter an amount greater than \$0.00"** and let them enter the amount again.
- Your program must use at least **one user-defined function** to make the calculations. And you can use the **math** for the built-in function **`math.floor()`** to separate dollars from cents
- The program must allow the user to enter the dollar and cents **as one input** (variable) – **not two different inputs**.

Here are some sample runs of the program:

```
Please enter the amount to make change for: $ 42.88
Number of hundreds:..... 0
Number of 50s :..... 0
Number of 20s:..... 2
Number of tens:..... 0
Number of fives:..... 0
Number of singles:..... 2
Number of quarters:..... 3
Number of dimes:..... 1
Number of nickels:..... 0
Number of pennies:..... 4
```

```
Please enter the amount to make change for: $ 1100.22
Number of hundreds:..... 11
Number of 50s :..... 0
Number of 20s:..... 0
Number of tens:..... 0
Number of fives:..... 0
Number of singles:..... 0
Number of quarters:..... 0
Number of dimes:..... 2
Number of nickels:..... 0
Number of pennies:..... 3
```

Program #2: Calculate the weekly pay (25 pts)

- Create a Python program named **test2-2.py** to calculate the weekly pay with or without overtime hours
- Your program should have at least one user-defined function to calculate Weekly Wages after **accepting** as the total Hours and hourly Wage and **return total pay**
- Your program will accept total hours worked in one week
- Then it will Calculate **the weekly pay with or without overtime hours**
- Use conditional statements (if...else...) to determine if the hours entered are more than 40 hours. If that is the case, then multiply any hour beyond **40 by 1.5 then add it to 40 hours then multiply it by the hourly wage** to come up with the total pay.
- Display a message if they have **worked overtime hours** or if **they did not have any overtime hours** (see the example of output (for both cases))

Example 1 with 5 overtime hours:

```
Enter hours worked: 45
Enter dollars paid per hour: 15
your overtime hours this week are: 5.0 hours
Wages for 45.0 hours at $15.00 per hour are $712.50.
```

Example 2 with No overtime hours:

```
Enter hours worked: 39
Enter dollars paid per hour: 15
you had no overtime hours this week
Wages for 39.0 hours at $15.00 per hour are $585.00.
```

Program #3: Two numbers comparison (15 pts)

- Create a Python program named `test2-3.py`
- In the main part of the program prompt the user to enter two integer numbers.
- Then, call a **user-defined function** and pass these two numbers into it.
- Use a Python **IF - ELSE IF - ELSE statement** to compare the two numbers and then display **one of three** messages:
 - The two numbers are equal.
 - The first number was larger than the second number
 - The second number was larger than the first number

(where or is the number the user typed in)

Below is a sample run of the program:

```
Please enter the first number: 33
Please enter the second number: 21
The first number 33 was larger than second number 21
```