**Projects for Chapter 04**

## 4-1 Project Statement: Create a program that defines the main() function and have it call another function to add two integer arguments. Pass two arguments to the function from main(). Have the function return the result to main() where it is then displayed on the console.

Calling a function to add two numbers

=======================================

Enter one number: XX

Enter another number: XX

Result: XX

## Operation

## define a main() function

## define a second function, addition()

## the addition() function should accept two arguments, num1 & num2

## the main() function should invoke addition() and pass two arguments to it

## return the result to main() and display the formatted result

## Specifications

## Assume the user will enter valid data.

## Save the file with the filename xxx\_addition.py (where xxx = your initials) or as defined by your instructor

## 4-2 Project Statement: Random numbers generated by a computer are not truly random. An algorithm is used to simulate generating random numbers and that algorithm provides surprising accuracy. Test this assumption by creating a program that acts to roll a 6-sided die 600 times. Count how many times each face of the die (1, 2, 3, 4, 5, or 6) occurs. If the numbers generated are random, you would expect each value to appear 100 times. The program should display four trial runs in tabular form, as shown below.

**Challenge:** Use both the random() function and the randint() for two separate displays and compare the accuracy (i.e., randomness) of each function. Is there a difference? Why or why not, in your opinion?

Rolling a die

=========================================

Trial number 1

Using the randint() function

1 was rolled xxx times.

2 was rolled xxx times.

3 was rolled xxx times.

4 was rolled xxx times.

5 was rolled xxx times.

6 was rolled xxx times.

=========================================

Trial number 2

Using the randint() function

1 was rolled xxx times.

2 was rolled xxx times.

3 was rolled xxx times.

...

...

=========================================

Trial number 1

Using the random() function

1 was rolled xxx times.

2 was rolled xxx times.

3 was rolled xxx times.

4 was rolled xxx times.

5 was rolled xxx times.

6 was rolled xxx times.

=========================================

Trial number 2

Using the random() function

1 was rolled xxx times.

2 was rolled xxx times.

3 was rolled xxx times.

...

...

## Operation

* import the standard random module
* main() should call a function that generates 600 random numbers between 1 and 6, inclusive and counts the number of times each of those six numbers is generated
* this function should be called four times from main()
* if both parts of this problem are done, there should be two functions: one using the random() function to generate the numbers and another using randint()
* to use the random() function to generate integer values from 1 to N use:

int(random.random() \* N) + 1, where N = the upper limit desired

* Display the results in tabular form

## Specifications

* Save the file with the filename xxx\_random\_test.py (where xxx = your initials) or as instructed by your instructor.

## 4-3 Project Statement: Create a program that converts a user’s entry from meters to feet or feet to meters. Import the application as a module into another application. The new application should ask the user which conversion will be performed, accept user input, and pass that value to the appropriate function. The result should be displayed and rounded. Include an option to run the application as often as desired. The final display should be as shown:

This program converts between feet and meters

Select from the two options below:

a. Feet to Meters

b. Meters to Feet

Which conversion would you like? (a/b): a

Enter feet: XX.XX

XX.XX meters

Would you like to convert another number? Enter (y/n): n

Thanks, bye!

## Operation

## the application will have two files, one which is imported as a module as 'cm'

## within the module there are two functions: to\_meter() and to\_feet()

## the application should contain the following functions: main(), fm\_welcome(), and fm\_menu()

## display a welcome message and menu

## allow the user to select which operation will be performed

## pass user input to the appropriate function

## calculate and display the results

## prompt the user to run the application again

## Specifications

* Assume the user will enter valid data.
* Save the files with the filenames xxx\_conversion\_module.py and xxx\_conversion\_application (where xxx = your initials) or as defined by your instructor.

## 4-4 Project Statement: Create a program that accepts the cost of items (as many as the user wants), one at a time and then calculates and displays the total cost of the purchase, the sales tax, and the cost to the user. The final display should be as shown:

Welcome to the 6% tax calculator!

Enter the cost of your items, one at a time.

first item: xx.xx

next item (enter 0 to end): xx.xx

next item (enter 0 to end): xx.xx

next item (enter 0 to end): 0

The sales tax on a total of $ xx.xx is $ xx.xx

The total amount after tax is: xx.xx

Would you like to calculate another? (y/n): y

## Operation

* the application will have two files, one which is imported as a module as tt
* the module should include purchases() and sales\_tax() functions
* the main() function should display a welcome message and transfer control to purchases(), where the user is prompted for input
* a running total of user input in purchases() is kept, used for a subtotal display, and passed as an argument to sales\_tax()
* the calculated sales tax and total are displayed

## Specifications

* Assume the user will enter valid data.
* Save the files with the filenames xxx\_totaltax\_module.py and xxx\_totaltax\_application.py (where xxx = your initials) or as defined by your instructor.