

Chapter 1 [branching, iteration, string and input, control structures]
Chapter 2 [functions, files, modules, recursion, system functions and parameters, global variables]
Chapter 3 [strings, list, tuple, dictionary, functions as object, lists and mutability]
Chapter 4 [exceptions, debugging, assertions]
Chapter 5 [classes, inheritance, encapsulation, abstract data types]
Chapter 6 [search algorithm, sort algorithm, hash tables]
Chapter 7 [re, pylab, networking, multithreading]
Chapter 8 [encryption-decryption, classic ciphers, turtle, tkinter]

Chapter 1

- 1) Create a calculator using conditional and loop statements. Calculator should print the operators available for use and check if user is entering one of those operators only. If an incorrect operator is used, program should print an error.
Code for addition, subtraction, multiplication and division.
- 2) Write a program that takes following two inputs from the user:
 - a) Gender
 - b) Current Salary
 1. If current salary is less than 1000 and gender is male, give a bonus of 5% to the user.
 2. If current salary is less than 1000 and gender is female, give a bonus of 5.5% to the user.

3. If current salary is more than 1000, give a bonus of 5% irrespective of the gender.
- 3) Make user input a number using input. Check and notify user if the number entered by user is prime or not.
- 4) Write a Python program which accepts the radius of a circle from the user and compute the area.
- 5) Write a Python program to print the following string in a specific format (see the output). *Sample String :*
- "Twinkle, twinkle, little star, How I wonder what you are! Up above the world so high, Like a diamond in the sky. Twinkle, twinkle, little star, How I wonder what you are" *Output :*

Twinkle, twinkle, little star,

How I wonder what you are!

Up above the world so high,

Like a diamond in the sky.

Twinkle, twinkle, little star,

How I wonder what you are

Chapter 2

- 1) Write a Python function that takes a list and returns a new list with unique elements of the first list.

Sample List : [1,2,3,3,3,3,4,5]

Unique List : [1, 2, 3, 4, 5]

- 2) Write a Python function that checks whether a passed string is palindrome or not.

Sample Input: MADAM, NURSES

- 3) Write a Python function to solve the Fibonacci sequence using recursion.

- 4) Write a Python program to copy the contents of a file to another file.

- 5) Write a Python function that accepts a string and calculate the number of uppercase letters and lowercase letters.

Sample String : 'The quick Brown Fox'

Expected Output :

No. of Uppercase characters : 3

No. of Lowercase Characters : 12

Chapter 3

- 1) Make user enter a planet name. If the planet entered by him is an inner planet, print that he chose an inner planet. Else print out that it's an outer planet. Also, check that user enters a planet name only and not something else. Upon encountering something other than a planet name, print a message asking user to provide acceptable input value. Make use of list.
- 2) Create a program with following lists where:
 - a) l1 is a list of integers from 1 to 10 (including 10).
 - b) l2 is a list of integers from 10 to 100 (including 100) where each element is in step of 10. Use range() function
 - c) l3 is a list of following strings:
 - i) Python
 - ii) Django
 - iii) Flask
 - iv) String
 - v) Function
 - vi) classes
 - d) l4 is a list of dictionaries where keys and values are following:
 - i) l1 → list l1
 - ii) l2 → list l2

iii) l3 → list l3

Note that dictionary keys need to be immutable. So l1, l2 and l3 keys should be used in string form and not as is. Whereas values should actually point to the lists l1, l2 and l3.

iv)

- e) Create a new list which has all the values in above created l1, l2 and l3 (created in the above program). Name the new list as “ main_list”.
- f) On list l1, perform following the operation l1 * 2 . Observe its result and store it in a new list called l5.
- g) Append l5 to main_list.
- h) Print occurrences of integer 1 in the main_list.

3) Concatenate following variables:

a) x = 9

b) y = “ is a square of “

c) z = 3

Beware that not all of the above variables are string.

4) Write a Python script to print a dictionary where the keys are numbers between 1 and 15 (both included) and the values are square of keys.

5) Write a Python program to get the 4th element and 4th element from last of a tuple.

Tuple: (13, 14, 15, 16, 17, 18, 19, 20)

Chapter 4

1) Write a program to prompt for a score between 0.0 and 1.0. If the score is out of range, print an error message. If the score is between 0.0 and 1.0, print a grade using the following table:

| Score | Grade |
|------------|-------|
| ≥ 0.9 | A |
| ≥ 0.8 | B |
| ≥ 0.7 | C |
| ≥ 0.6 | D |
| < 0.6 | F |

```
Enter score: 0.95
A
```

```
Enter score: perfect
Bad score
```

```
Enter score: 10.0
Bad score
```

```
Enter score: 0.75
C
```

```
Enter score: 0.5
F
```

2) Find potential sources of logic errors in this code snippet:

```
product = 0
for i in range(10):
    product *= i

sum_squares = 0
for i in range(10):
    i_sq = i**2
    sum_squares += i_sq

nums = 0
for num in range(10):
    num += num
```

3) This function adds an element to a list inside a dict of lists. Rewrite it to use a *try-except* statement which handles a possible `KeyError` if the list with the name provided doesn't exist in the dictionary yet, instead of checking beforehand whether it does. Include *else* and *finally* clauses in your *try-except* block:

```
def add_to_list_in_dict(thedict, listname, element):
    if listname in thedict:
        l = thedict[listname]
        print("%s already has %d elements." % (listname, len(l)))
    else:
        thedict[listname] = []
        print("Created %s." % listname)

    thedict[listname].append(element)

    print("Added %s to %s." % (element, listname))
```

4) Write a python program to illustrate `ZeroDivisionError`.

5) Write a python program which takes age as an input. Age must be a positive or zero. If age is invalid, raise a `ValueError`.

Chapter 5

- 1) Write a Python class which has two methods `get_String` and `print_String`. `get_String` accept a string from the user and `print_String` print the string in uppercase.
- 2) Write a Python class named `Circle` constructed by a radius and two methods which will compute the area and the perimeter of a circle.
- 3) Write a simple program which loops over a list of user data (tuples containing a username, email and age) and adds each user to a directory if the user is at least 16 years old. You do not need to store the age. Write a simple exception hierarchy which defines a different exception for each of these error conditions:
 1. the username is not unique
 2. the age is not a positive integer
 3. the user is under 16
- 4) Create a base `Polygon` class. This class has data attributes to store the number of sides, `n` and magnitude of each side as a list, `sides`. Method `inputSides()` takes in magnitude of each side and similarly, `dispSides()` will display these properly. Create an inherited class `Triangle`. class `Triangle` has

a new method findArea() to find and print the area of the triangle.

- 5) Write a Python class which stores information about a person like [first_name, last_name, titles, email]. Define a class attribute TITLES = ['Mr', 'Mrs', 'Ms', 'Dr']. If users title is not in TITLES, raise a ValueError.

Chapter 6

- 1) Complete the following code which will perform a selection sort in Python. "... " denotes missing code that should be filled in:

```
def selection_sort(items):  
    """Sorts a list of items into ascending order using the  
    selection sort algorithm.  
    """  
    for step in range(len(items)):  
        # Find the location of the smallest element in  
        # items[step:].  
        location_of_smallest = step  
        for location in range(step, len(items)):  
            # TODO: determine location of smallest  
            ...  
        # TODO: Exchange items[step] with items[location_of_smallest]  
        ...
```

- 2) Write a function which implements linear search. It should take a list and an element as a parameter, and return the position of the element in the list. If the element is not in the list, the function should raise an

exception. If the element is in the list multiple times, the function should return the first position.

- 3) Using two Arrays of equal length, create a Hash object where the elements from one array (the keys) are associated with the elements of the other (the values):

Sample Input:

```
keys = ['a', 'b', 'c']
```

```
values = [1, 2, 3]
```

Sample Output:

```
{'a': 1, 'c': 3, 'b': 2}
```

- 4) Write a python program to hash the string: “Hello Monty Python” with SHA256.
- 5) Write a Python program for binary search.

Binary Search : In computer science, a binary search or half-interval search algorithm finds the position of a target value within a sorted array. The binary search algorithm can be classified as a dichotomies divide-and-conquer search algorithm and executes in logarithmic time.

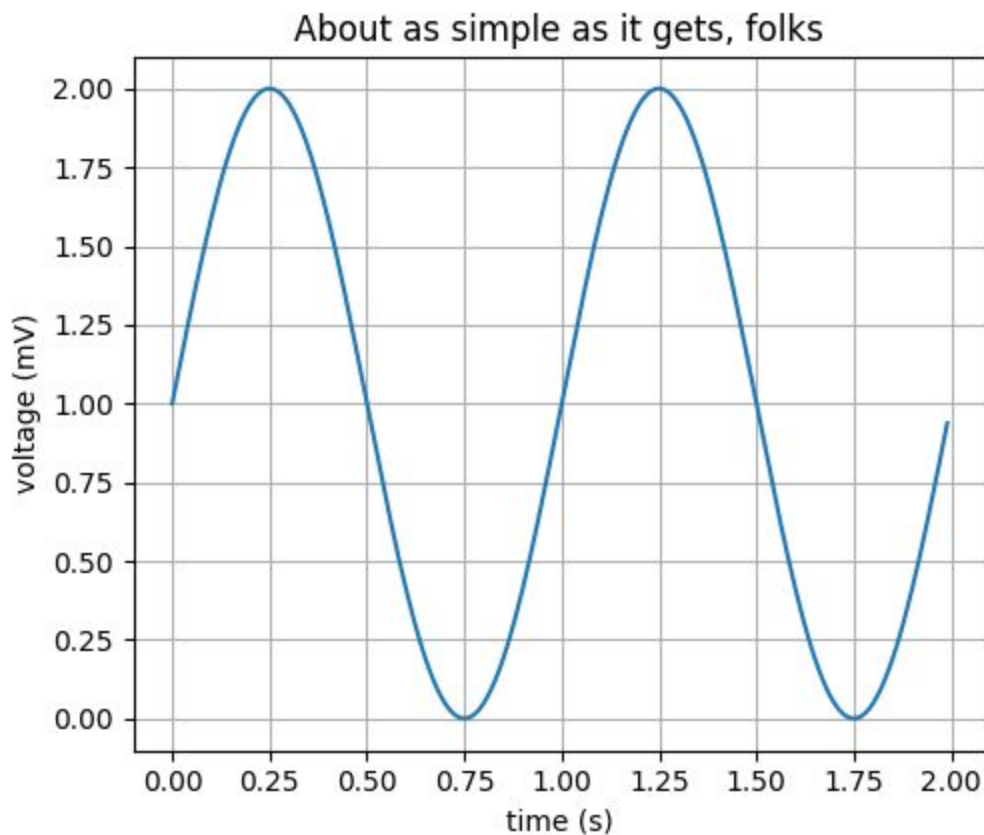
Test Data :

```
binary_search([1,2,3,5,8], 6) -> False
```

```
binary_search([1,2,3,5,8], 5) -> True
```

Chapter 7

- 1) Write a Python program to check that a string contains only a certain set of characters (in this case a-z, A-Z and 0-9).
- 2) Write a Python program that matches a string that has an a followed by one or more b's.
- 3) Create socket with `socket.socket` function.
- 4) Write a python program which creates a new thread after every 5 seconds.
- 5) Write a simple python pylab program to print this:



Chapter 8

- 1) Create a simple python program with tkinter, which has two buttons on it *Greet* and *Close*. When user clicks on *Greet*, print “Greetings!” in the console, and when user clicks on *Close*, close the program.
- 2) Write a python program to implement monoalphabetic cipher.
- 3) Write a simple python program to encode a string “Hello monty python” with “base64”
- 4) Write a simple tkinter program which converts feet to meters.
- 5) Write a simple turtle program to print this:

