



G. H. Raisoni  
College of Engineering & Management

Wagholi, Pune

# Certificate

*This is to certify that*

*Mr./Ms. Pratik Rajesh Jade*

*studying in Semester/Year First Year*

*Branch Artificial Intelligence*

*Name of the Dept. F.Y. B. Tech*

*Section A*

*has satisfactorily completed the practical work of*

*subject Programming for Problem Solving*

*during the academic year 2021*

\_\_\_\_\_  
*Signature of Subject Teacher*

\_\_\_\_\_  
*Signature of Head of Department*



## INDEX PAGE



**RAISONI GROUP**  
— A vision beyond —

Name - Pratik Rajesh Jade  
 Roll no. - A72

Experiment no: 01

Aim: Implementing if else in Python

Problem Statement: Write a python program for checking whether the given number is an even number or not.

Theory: A number is even if it is perfectly divisible by 2. When the number is divided by 2, we use the remainder operator % to compute the remainder. If the remainder is not zero, the number is odd.

Algorithm:

1. Take number as an input

2. Mod = num % 2

3. If mod > 0

Print odd number

else

Print even number

Python code:

input:→ num = int(input("Enter a number:"))

mod = num % 2

If mod > 0

print("This is an odd number:")

else

print("This is an even number:")

Output:-

Enter a number: 10

This is an even number

| M         | T     | W | T | F | S | S |
|-----------|-------|---|---|---|---|---|
| Page No.: | YOUVA |   |   |   |   |   |
| Date:     |       |   |   |   |   |   |

Name: Pratik Rajesh Jade  
Roll no: A72

## Experiment No: 02

Problem statement: Write a python program using While loop to print 10 to 1 number.

Theory: The while loop in python is used to iterate over a block of code as long as the test expression (condition) is true.  
We generally use the loop when we don't know beforehand, the number of times to iterate

Syntax of while loop in python:

While test expression:

Body of while

In while loop, test expression is checked first. The body of the loop is entered only if the test expression evaluates to True. After one iteration, the test expression is checked again. This process continues until the test expression evaluates to False.

In python, the body of the while loop is determined through indentation.

Body starts with indentation and the first unindented line marks the end.

Python interprets any non-zero value as True. None and 0 are interpreted as false.

Algorithm:

1. initialise value of i
2. put the condition as `while(i >= 1):`
3. print i
4. Decrement value of i

Python code:

```
input- i = 10;
while (i >= 1):
    print (i)
    i = i - 1
```

Output- 10, 9, 8, 7, 6, 5, 4, 3, 2, 1

Problem Statement: 2.2: Write a python program to print a table of any numbers using for loop.

Theory:

What is for loop in python?

The for loop in Python is used to iterate over a sequence (list, tuple, string) or other iterable object. Iterating over a sequence is called traversal.

Syntax of for loop

for val in sequence:

Body of for

Here, val is the variable that takes the value of the item the sequence on each iteration.

Loop continues until we reach the last item in the sequence. The body of the for loop is separated from the rest of the code using indentation.

The range() function

We can generate a sequence of number using range() function. range(10) will generate number from 0 to 9 (10 numbers).

We can also define the start, stop & step size as range(start, stop, step, size). Step size defaults to 1 if not provided.

This function does not store all the values in memory, it would be inefficient. So, it remembers the start, stop, step size and generates the next number on the go. To force this function to output all the items, we can use the function list().

Algorithm:

1. Take input from user
2. Apply specific range
3. print

Python code:

```

digits = [0, 1, 5]
for i in digits:
    print(i)
else:
    print("No items left.")

```

Output

0  
1

5

No items left

Name - Pratik Rajesh Jade  
Rollno - A72

Experiment No: 03

Problem Statement: Write a python program to demonstrate a concept of a function.

Theory: What is a function in python?

In Python, function is a group of related statements that perform a specific task. Functions help break our program into smaller and modular chunks. As our program grows larger and larger, functions make it more organized and manageable. Furthermore, it avoids repetition & makes code reusable.

Syntax of function:

```
def function_name(parameters):
    """
    do something
    statement(s)
```

Above shown is a function definition which consists of following components.

1. Keyword `def` marks the start of function header.
2. A function name to uniquely identify it. Function naming follows the same rules of writing identifiers in Python.
3. Parameters (arguments) through which we pass values to a function. They are optional.
4. A colon (`:`) to mark the end of function header.
5. Optional documentation string (docstring) to describe what the function does.
6. One or more valid python statements that make up the function body. Statements must have same indentation level (usually 4 spaces).
7. An optional return statement to return a value from the function.

Python code :

```
def absolute_value(num):
    """ This function return the absolute
    value of the entered number """
    if num >= 0:
        return num
    else:
        return -num
print(absolute_value(2))
print(absolute_value(-4))
```

Output:

2

4

Name - Pratik Rajesh Jade  
 Rollno - A72

### Experiment - 04

Problem Statement : Implementing Tuple & Dictionary in python

Theory : Tuple

A tuple is same as list, except that the set of elements is enclosed in parentheses instead of square brackets.

A tuple is an immutable list. i.e once a tuple has been created, you can't add elements to a tuple or remove elements from the tuple.

Benefit of Tuple:

Tuples are faster than lists.

1. If the user wants to protect the data from accidental changes, tuple can be used.
2. Tuple can be used as keys in dictionaries while list can't.

Program :

```
my_tuple = (1, 2, 3)
print (my_tuple)
```

Output :

(1, 2, 3)

Dictionaries:

lists are ordered sets of objects, whereas dictionaries are unordered sets

1. Dictionary is created by using curly brackets i.e. {}
2. Dictionaries are accessed via keys & not via their position.
3. A dictionary is an associative array (also known as hash). Any key of the dictionary is associated (or mapped) to a value.

Input:

```
my_dict = {'name': 'Jack', 'age': 26}  
print(my_dict['name'])
```

Output: Jack

Name - Pratik Jade  
Rollno - A72

### Experiment no - 5

Problem Statement : Algorithm, Expression, Variables & I/o

#### Theory :

#### Algorithm

It is defined as a sequence of instructions that describe a method for solving a problem. In other words it is a step by step procedure for solving a problem.

#### Properties of Algorithms

- 1) Each and every instruction should be precise and unambiguous.
- 2) Instructions in a algorithm should not be repeated infinitely.
- 3) Algorithm should conclude after a finite number of steps.
- 4) Should have an end point
- 5) Desired result should be obtained only after the algorithm terminates.

#### Example

Write an algorithm to print "Good Morning"

Step 1: Start

Step 2: Print "Good Morning"

Step 3: Stop.

#### Variables:

A variable allows us to store a value by assigning it to a name which can be used later.

Named memory locations to store values. Programmers generally choose names for their variables that are meaningful. It can be of any length. No space is allowed.

enter the number 3

# python accepts string as default data type, conversion is required for type.

Output: Output can be displayed to the user using print statement.

Syntax:

- print (expression/constant/Variable)

Example:

print ("Hello")Hello

| M         | T     | W | T | F | S | S |
|-----------|-------|---|---|---|---|---|
| Page No.: | YOUVA |   |   |   |   |   |
| Date:     |       |   |   |   |   |   |

Name - Pratik Jade  
Rollno - A72

## Experiment No - 06

### Problem Statement : Control Structures

Theory: Control structures are a flowchart method to represent the execution flow of programming languages.

#### Conditionals:

Conditional if

Alternative if...else

Chained if ... elif ... else

Nested if ... else

Conditional (if) is used to test a condition, if the condition is true the statements inside if will be executed

#### Syntax:

if (condition):

#### Statement 1

alternative (if-else):

In the alternative the condition must be true or false. In this else statement can be combined with if statement. The else statement contains the block of code the executes when the condition is false. If the condition is true statement inside the if get executed otherwise else part gets executed. The alternatives are called branches, because they are branches in the flow of execution.

## Syntax:

`If (condition 1):`

`Statement 1`

`else`

`Statement 2`

## Chained conditionals (if - elif - else)

- The `elif` is short for `else if`.
- This is used to check more than one condition.
- If the condition 1 is false, it checks the condition 2 of the `elif` block.

If all the conditions are false, then the `else` part is executed.

Among the several `if - elif - else` part, only one part is executed according to the condition.

- The `if` block can have only one `else` block. But it can have multiple `elif` blocks.
- The way to express a computation like that is chained conditional.

## Syntax:

`If (condition 1):`

`Statement 1`

`elif (condition 2):`

`Statement 2`

`elif (condition 3):`

`Statement 3`

`else:`

`default statement`

## Nested conditionals

One conditional can also be nested within another. Any number of condition can be nested inside one another. In this, if the condition is true it checks another if condition 1. If both the conditions are true statement 1 get executed otherwise statement 2 get execute. If the condition is false statement

3 get executed

Syntax:

```
If (Condition):
    if (Condition):
        Statement 1
    else:
        Statement 2
else:
    Statement 3
```

Iteration / Control statements:

For loop:

for in range:

We can generate a sequence of numbers using range() function. range(10) will generate numbers from 0 to 9 (10 numbers).

In range function have to define the start, stop and step size as range(start, stop, step, size). Step size defaults to 1 if not provided.

Syntax

```
for i in range(Start, stop, steps):
    body of for loop.
```

While loop

While loop statement in Python is used to repeatedly executes set of statement as long as a given condition is true.

In while loop, test expression is checked first. The body of the loop is entered only if the test expression is True. After one iteration, the test expression is checked again. This process continues until the test-expression evaluates to false.

In python, the body of the while loop is determined through indentation. The statements inside the starts with indentation & the first unindented line marks the end.

### Syntax.

initial value

while (condition):

body of while loop

increment

### Programme :

i = 1

while i < 6:

    point(i)

    i += 1

Output → 1

2

3

4

5

| M         | T     | W | T | F | S | S |
|-----------|-------|---|---|---|---|---|
| Page No.: | YOUVA |   |   |   |   |   |
| Date:     |       |   |   |   |   |   |

Name - Pratik Jade  
Roll no - A72

Experiment no: 07

Problem Statement : List, Strings, Tuples & Dictionary.

### Theory

#### List

1) List. List are used to store multiple items in a single variable.

Lists are one of 4 built-in data types in Python used.

2) List items. List items are ordered, changeable, and allow duplicate value.

List items are indexed, the first item has..

3) Ordered. When we say that lists are ordered, it means that the items have a defined order, and that order will not.

4) Changeable. The list is changeable, meaning that we can change, add, and remove items in a list after it has been...

#### Programme:

```
thislist = ["apple", "banana", "cherry"]
print(thislist)
```

Output: ['apple', 'banana', 'cherry']

#### String

In python strings are arrays of bytes representing Unicode characters. However, Python does not have a character data type, a single character is simple a string with a length of 1. Square brackets can be used to access elements of the strings. Strings in Python can be created using single quotes or double quotes or even triple quotes.

## Programme:

```
a = "Hello"
```

```
print(a)
```

Output: Hello

## Tuple

A tuple is same as list, except that the set of elements is enclosed in parentheses instead of square brackets. A tuple is an immutable list, i.e.

Once a tuple has been created, you can't add elements to a tuple or remove element from the tuple.

## Benefit of Tuple:

Tuples are faster than list.

1. If the user wants to protect the data from accidental changes, tuple can be used.
2. Tuples can be used as keys in dictionaries, while list can't.

## Program:

```
my_tuple = (1, 2, 3)
```

```
print(my_tuple)
```

Output: (1, 2, 3)

## Dictionaries

Lists are ordered sets of objects, whereas dictionaries are unordered sets.

1. Dictionary is created by using curly brackets, i.e {}.
2. Dictionaries are accessed via keys & not via their position.
3. A dictionary is an associative array (also known as hashes). Any key of the dictionary is associated (or mapped) to a value.

| M         | T     | W | T | F | S | S |
|-----------|-------|---|---|---|---|---|
| Page No.: | YOUVA |   |   |   |   |   |
| Date:     |       |   |   |   |   |   |

Programme:

```
my_dict = {'name': 'Jack', 'age': 26}
print(my_dict['name'])
```

Output : Jack

Name - Pratik Jade  
Rollno - A72

Experiment no : 08

### Problem Statement : Function

#### Theory:

#### Functions:

Function is a sub program which consists of set of instructions used to perform a specific task. A large program is divided into basic building blocks called function.

#### Need for Function:

- When the program is too complex and large they are divided into parts. Each part is separately coded and combined into single program. Each subprogram is called as function.
- Debugging Testing and maintenance becomes easy when the program is divided into subprograms.
- Functions are used to avoid rewriting same code again & again in a program.
- Function provides code re-usability
- The length of the program is reduced.

#### Types of function:

Functions can be classified into two categories:

- 1) User defined function
- 2) Built in function

#### Programme:

```
def multiply(x,y):    ← Declaring arguments
    print(x+y)
```

```
multiply(2,8) ← Passing arguments.
```

Output : 16

| M         | T     | W | T | F | S | S |
|-----------|-------|---|---|---|---|---|
| Page No.: | YOUVA |   |   |   |   |   |
| Date:     |       |   |   |   |   |   |

Name - Pratik Jade  
Rollno - A72

Experiment no : 09

Problem Statement : Object oriented programming.

Theory:

Python is a multi-paradigm programming language. It supports different programming approaches.

One of the popular approaches to solve a programming problem is by creating objects. This is known as Object-Oriented Programming (OOP).

An object has two characteristics:

- Attributes
- behavior

Let's take an example:

A parrot is an object, as it has the following properties:

- name, age, color as attributes
- Singing, dancing as behavior

The concept of OOP in Python focuses on creating reusable code. This concept is also known as DRY (Don't Repeat Yourself).

In Python, the concept of OOP follows some basic principles:

Class

A class is a blueprint for the object.

We can think of class as a sketch of a parrot with labels. It contains all the details about the name, colors, size, etc. Based on these descriptions, we can study about the parrot. Here, a parrot is an object.

The example of class of parrot can be:

Class Parrot:

pass

Here, we use the class keyword to define an empty class Parrot. From class, we construct instances. An instance is a specific object created from a particular class.

Object

An object(instance) is an instantiation of a class. When class is defined, only the ~~class~~ description for the object is defined. Therefore, no memory or storage is allocated.

The example for object of parrot class can be:

Obj = Parrot()

Here, Obj is an object of class Parrot.

Suppose we have details of parrots. Now, we are going to show how to build the class and objects of parrots.

Programme:

class Person:

def \_\_init\_\_(self, name, age):

self.name = name

self.age = age

p1 = Person("John", 36)

print(p1.name)

print(p1.age)

Output: John

36

| M         | T     | W | T | F | S | S |
|-----------|-------|---|---|---|---|---|
| Page No.: |       |   |   |   |   |   |
| Date:     | YOUVA |   |   |   |   |   |

Name - Pratik Jade

Rollno - A72

Experiment No: 10

Problem Statement : Write a python program to read a file and display content in a file.

Theory:

Python File Open:

File handling is an important part of any web application.

Python has several functions for creating, reading, updating and deleting files - File Handling

The key function for working with files in Python is the `open()` function.

The `open()` function takes two parameters; filename, and mode.

There are four different method (modes) for opening a file:

'r' - Read - Default value. Opens a file for reading, error if the file does not exist.

'a' - Append - Opens a file for appending, creates the file if it does not exist.

'w' - Write - Opens a file for writing, creates the file if it does not exist

'x' - Create - Create the specified file, returns an error if the file exists

In addition you can specify if the file should be handled as binary or text mode

't' - Text - Default value . Text mode

'b' - Binary - Binary mode (eg- images)

Syntax

To open a file for reading it is enough to specify the name of the file:

`F = open ("demofile.txt")`

The code above is the same as:

`F = open ("demofile.txt", "rt")`

Example : Create text file

demofile.txt

Hello ! Welcome to demofile.txt

This file is for testing purposes.

Good luck !

To open the file, use the built-in `open()` function.

The `open()` function return a file object, which has `read()` method for reading the content of the file:

Example :

```
F = open("demofile.txt", "r")
print(F.read())
```

Output :

Hello ! Welcome to demofile.txt

This file is for testing purposes.

Good luck !

Programme :

```
from google.colab import files
with open('example.txt', 'a') as F
    F.write('Welcome to GHRCEM PUNE')
files.download('example.txt')
```

Output :

Welcome to GHRCEM PUNE.