

INDEX

Sl. No.	Name Of The Experiment	PAGE NO
1	Introduction to Python	07
2	Write instructions to perform each of the steps below (a) Create a string containing at least five words and store it in a variable. (b) Print out the string. (c) Convert the string to a list of words using the string split method. (d) Sort the list into reverse alphabetical order using some of the list methods (you might need to use dir(list) or help(list) to find appropriate methods). (e) Print out the sorted, reversed list of words.	09
3	Write a program that determines whether the number is prime.	10
4	Find all numbers which are multiple of 17, but not the multiple of 5, between 2000 and 2500?	11
5	Swap two integer numbers using a temporary variable. Repeat the exercise using the code format: a, b = b, a. Verify your results in both the cases.	12
6	Find the largest of n numbers, using a user-defined function largest().	13
7	Write a function my Reverse() which receives a string as an input and returns the reverse of the string.	14
8	Check if a given string is palindrome or not.	15
9	WAP to convert Celsius to Fahrenheit.	16
10	Find the ASCII value of characters.	17
11	WAP for simple calculator.	18

SYLLABUS

PR-3 PYTHON PROGRAMMING LAB

Total Periods	60	Maximum Marks	75 Marks
Lab. Periods:	4 Periods /week	Term Works	25 Marks
Examination	3hours	End Semester Examination	50Marks

RATIONALE

This course introduces to the students the Python language. Upon completion of this course, the student will be able to write non trivial Python programs dealing with a wide variety of subject matter domains. Topics include language components, the IDLE/IDE environment, control flow constructs, strings, I/O, collections, classes, modules, and regular expressions

LEARNING OUTCOMES

After undergoing the course, the students will be able to:

- Execute Python code in a variety of environments
- Use correct Python syntax in Python programs
- Use the correct Python control flow construct
- Write Python programs using various collection data types
- Write home grown Python functions
- Use many of the standard Python modules such as os, sys, math, and time
- Trap various errors via the Python Exception Handling model
- Use the IO model in Python to read and write disk files
- Create their own classes and use existing Python classes. Understand and use the Object Oriented paradigm in Python programs
- Use the Python Regular Expression capabilities for data verification

DETAILED CONTENTS

1. Introduction
 - Brief History of Python
 - Python Versions
 - Installing Python
 - Environment Variables
 - Executing Python from the Command Line
 - IDLE
 - Editing Python Files
 - Python Documentation
 - Getting Help
 - Dynamic Types
 - Python Reserved Words
 - Naming Conventions

2. Basic Python Syntax
 - Basic Syntax
 - Comments
 - String Values
 - String Methods
 - The format Method
 - String Operators
 - Numeric Data Types
 - Conversion Functions
 - Simple Output
 - Simple Input
 - The % Method
 - The print Function
3. Language Components
 - Indenting Requirements
 - The if Statement
 - Relational and Logical Operators
 - Bit Wise Operators
 - The while Loop
 - break and continue
 - The for Loop
4. Collections
 - Introduction
 - Lists
 - Tuples
 - Sets
 - Dictionaries
 - Sorting Dictionaries
 - Copying Collections
 - Summary
5. Functions
 - Introduction
 - Defining Your Own Functions
 - Parameters
 - Function Documentation
 - Keyword and Optional Parameters
 - Passing Collections to a Function
 - Variable Number of Arguments
 - Scope
 - Functions - "First Class Citizens"
 - Passing Functions to a Function
 - map
 - filter
 - Mapping Functions in a Dictionary
 - Lambda
 - Inner Functions
 - Closures

6. Modules
 - Modules
 - Standard Modules - sys
 - Standard Modules - math
 - Standard Modules - time
 - The dir Function
7. Exceptions
 - Errors
 - Runtime Errors
 - The Exception Model
 - Exception Hierarchy
 - Handling Multiple Exceptions
 - Raise
 - assert
8. Input and Output
 - Introduction
 - Data Streams
 - Creating Your Own Data Streams
 - Access Modes
 - Writing Data to a File
 - Reading Data From a File
 - Additional File Methods
 - Using Pipes as Data Streams
 - Handling IO Exceptions
9. Classes in Python
 - Classes in Python
 - Principles of Object Orientation
 - Creating Classes
 - Instance Methods
 - File Organization
 - Special Methods
 - Class Variables
 - Inheritance
 - Polymorphism
10. Regular Expressions
 - Introduction
 - Simple Character Matches
 - Special Characters
 - Character Classes
 - Quantifiers
 - The Dot Character
 - Greedy Matches
 - Grouping
 - Matching at Beginning or End
 - Match Objects
 - Substituting
 - Splitting a String
 - Compiling Regular Expressions

LIST OF PRACTICALS

1. Write instructions to perform each of the steps below
 - (a) Create a string containing at least five words and store it in a variable.
 - (b) Print out the string.
 - (c) Convert the string to a list of words using the string split method.
 - (d) Sort the list into reverse alphabetical order using some of the list methods (you might need to use `dir(list)` or `help(list)` to find appropriate methods).
 - (e) Print out the sorted, reversed list of words.
2. Write a program that determines whether the number is prime.
3. Find all numbers which are multiple of 17, but not the multiple of 5, between 2000 and 2500?
4. Swap two integer numbers using a temporary variable. Repeat the exercise using the code format: `a, b = b, a`. Verify your results in both the cases.
5. Find the largest of n numbers, using a user defined function `largest()`.
6. Write a function `my Reverse()` which receives a string as an input and returns the reverse of the string.
7. Check if a given string is palindrome or not.
8. WAP to convert Celsius to Fahrenheit
9. Find the ASCII value of charades
10. WAP for simple calculator

Methodology

The Students shall be taught about the Syntax of Python Language which is similar to other High level languages in the initial 8 to 10 classes. Then after learning the syntax the students shall Write the codes for the Practical Exercise and Test its results in the Lab.

RECOMMENDED BOOKS

Sl.No	Name of Authors	Title of the Book	Name of the publisher
1	C Satyanarayan M, Radhika Mani, B N Jagdesh	Python Programming	University Press
2	Mark Lutz;	Learning Python by Mark Lutz;	Pratham Books, Bangalore
3	Robert Richards	Python Programming For Beginners: A Must Read Introduction to Python Programming	Pratham Books, Bangalore

PYTHON

Introduction :

- Python is a popular programming language.
- It combines the features of C and Java.
- It was created by Guido van Rossum, and released in 1991.
- The name "Python" was adopted from the Rossum's favourite comedy series "Monty Python's Flying Circus".
- Python is mainly interpreted language.
- It is an open source software.
- It is used for web development (server-side), software development, mathematics, system scripting etc.
- In 1994, Python 1.0 was released with new features like lambda, map, filter, and reduce.
- Python 2.0 added new features such as list comprehensions, garbage collection systems.
- On December 3, 2008, Python 3.0 (also called "Py3K") was released. It was designed to rectify the fundamental flaw of the language.
- ABC programming language is said to be the predecessor of Python language, which was capable of Exception Handling and interfacing with the Amoeba Operating System.

Application of Python :

- General purpose language: Used to create Machine learning, Web applications/development, GUI, Software development.
- Used alongside software to create workflows.
- Connect to database systems. It can also read and modify files.
- Can be used to handle big data and perform complex mathematics.
- Can be used for rapid prototyping, or for production-ready software development.
- Top companies using Python: Google, Dropbox, Youtube, Quora, Yahoo, NASA, Reddit

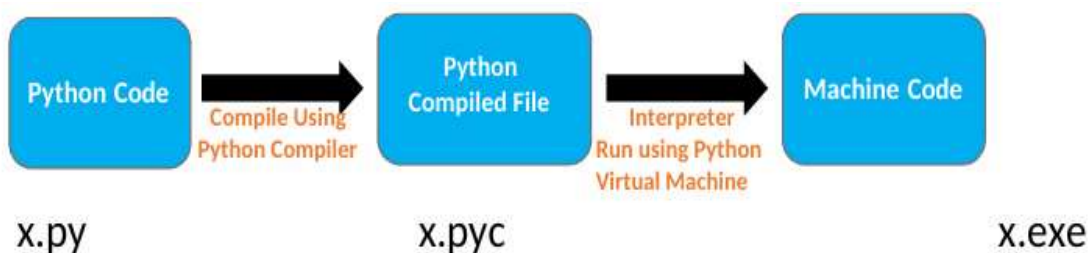


Byte Code :

- Python is usually called an interpreted language, however, it combines compiling and interpreting.
- When we execute a source code (a file with a .py extension),
- Python first compiles it into a byte code.
- The byte code is a low-level platform-independent representation of your source code, however, it is not the binary machine code and cannot be run by the target machine directly. In fact, it is a set of instructions for a virtual machine which is called the Python Virtual Machine (PVM).
- After compilation, the byte code is sent for execution to the PVM.
- The PVM is an interpreter that runs the byte code and is part of the Python system.
- The byte code is platform-independent, but PVM is specific to the target machine.
- The default implementation of the Python programming language is CPython which is written in the C programming language.
- CPython compiles the python source code into the byte code, and this bytecode is then executed by the CPython virtual machine.

Execution of a python program :

- Python file first get compiled to give us byte code and that byte code is interpreted into machine language.
- This is performed by PVM.
- **Execution:**



EXPERIMENT- 1

AIM OF THE EXPERIMENT :

Introduction to Python.

What is python?

Python is the popular programming language .It was created by Guido van Rossum sreleased in 1991.

It was used for

- Web development
- Software development
- Mathematics
- System scripting

What can python do?

- Python can be used on a server to create web page application.
- It is used alongside software to create work Flows.
- It can connect to database system. It can also read & modify.
- It can be used to handle big data & perform complex math.
- It can be used for rapid prototyping for producing.
- Ready software development

Why python?

- Python works on different platform (windows, Mac, Linux)
- It has simple syntax that allows developers to write program with fewer lines than some other programming language
- It runs on an interpreter system means that code can be executed as soon as it is written.
- It can be treated in a procedural way; an object oriented way are functional way.

ADVANTAGES:-

User friendly:-python's syntax is simple. It is much like writing in English, so it is easy to read, write & understand code. It makes developers more productive unlike java; you can get things done in less links of code.

Open sources community: - It is an open source & its source code is available to the public for free. You don't pay to use python one greatest things about python is its longer community.

Prototyping-Because of python has simple syntax reusability, developers are more Productive & can perform Fast prototyping. This give us the proof of concept & project Management. It also gives minimum variable product.

Portable-code we write on windows can be transformed to Mac as it is & the other way around.

High level- As we know python is a high-level language for programmers, there is no need to remember system architecture or manage memory. It also supports automatic garbage Collection.

Interpreted- Internally the source code converts to bit code. Python code executes lines & so easier to debug. Interpreting does make it slower as compared to java.

Object oriented-python is an object oriented language & also supports functional programming. It can model the real world It also supports the procedure oriented.

Dynamic Typing- Java is statically typed & python is dynamically typed. As a programmer we don't need to declare a data type for a value in advance, this is decided runtime.

Extensible Embeddable: - python is both extensible & embeddable. You can borrow code from other language to extend python's also embed that code.

DISADVANTAGES-

Slow speed:-We discuss above that python is an interpreted Language & dynamically typed language. These two features make the sport.

Not memory efficient:- If usage a large amount of memory, which is a disadvantage while building applications which need memory.

APPLICATION

- scripting
- Data science
- Machine language
- Artificial Intelligent
- GUI & game development
- Web development
- Complex vision / Image manipulation
- Rebooting
- Web scraping
- Network programming
- Numeric & specific computing
- Enterprise application

CONCLUSION:

From the above experiment we concluded that the experiment has successfully done.

EXPERIMENT - 2

AIM OF THE EXPERIMENT :

- (a) Create a string containing at least five words & store it in a variable.
- (b) Print out the string
- (c) Convert the string to a list of words.
- (d) Sort the list into reverse alphabetical order. Using some of the list methods.
- (e) Print out the sorted, reversed list of words.

CODE :

```
a = 'Subhasmita parida'
b = list(a)
print("Original list:", b)
b.sort()
print("Sorted list:", b)
b.sort(reverse=True)
print("Reverse sorted list:", b)
```

OUTPUT :

```
Subhasmita parida

['Subhasmita', 'Parida']

['Parida', 'Subhasmita']

['Subhasmita', 'Parida']
```

CONCLUSION:

From the above experiment we conclude that the experiment has successfully done.

EXPERIMENT - 3

AIM OF THE EXPERIMENT :

Write a program that determines whether the number is prime .

CODE :

```
number = int(input("Enter any number: "))
// prime number is always greater than 1
if number > 1:
    for i in range(2, number):
        if (number % i) == 0:
            print(number, "is not a prime number")
            break
    else:
        print(number, "is a prime number")

// if the entered number is less than or equal to 1
// then it is not prime number
else:
    print(number, "is not a prime number")
```

OUTPUT :

```
Enter no. 11
11 is a prime no.
```

CONCLUSION:

From the above experiment we concluded that the experiment has done successfully.

EXPERIMENT- 4

AIM OF THE EXPERIMENT :

Find all numbers which are multiple of 17 but not the multiple of 5,between 2000 and 2500.

CODE :

```
for i in range(2000, 2500):  
    if (i % 17 == 0) and (i % 5 != 0):  
        print(" ", i)
```

OUTPUT :

```
2006  
2023  
2057  
2074  
2091  
2108  
2142  
2159  
2176  
2193  
2227  
2244  
2261  
2278  
2312  
2329  
2346  
2363  
2397  
2414  
2431  
2448  
2482  
2499
```

CONCLUSION:

From the above experiment we conclude that the experiment has done successfully

EXPERIMENT- 5

AIM OF THE EXPERIMENT :

Swap two integer numbers using a temporary variable repeat the exercise using the code format
a,b=b,a

CODE :

```
# Initial values
```

```
a = 20
```

```
b = 30
```

```
temp = a
```

```
a = b
```

```
b = temp
```

```
print("a =", a)
```

```
print("b =", b)
```

```
a, b = b, a
```

```
print("a =", a)
```

```
print("b =", b)
```

OUTPUT :

```
a=20,b=30
```

CONCLUSION:

From the above experiment we concluded that the experiment has done successfully.

EXPERIMENT- 6

AIM OF THE EXPERIMENT :

Find the largest of n numbers, using a user defined function largest()

CODE :

```
def largest(lst):  
    max_val = lst[0]  
    for i in lst:  
        if i > max_val:  
            max_val = i  
    return max_val  
  
a = [10, 50, 2, 18, 3, 33]  
  
print("Largest number is:", largest(a))
```

OUTPUT :

largest number is;50

CONCLUSION:

From the above experiment we conclude that the experiment has done successfully.

EXPERIMENT- 7

AIM OF THE EXPERIMENT :

Write a function my Reverse() which receives a string as i/p & returns the reverse of string.

CODE :

```
def my_reverse(a):  
    b = a[::-1]  
    print(b)  
my_reverse("program")
```

OUTPUT :

margorp

CONCLUSION:

From the above experiment we conclude that the experiment has done successfully.

EXPERIMENT- 8

AIM OF THE EXPERIMENT :

Check if a given string is palindrome or not.

CODE :

```
s = 'pop'

if s == s[::-1]:
    print("Palindrome")
else:
    print("Not palindrome")
```

OUTPUT :

Palindrome

CONCLUSION:

From the above experiment we conclude that the experiment has been done successfully.

EXPERIMENT- 9

AIM OF THE EXPERIMENT :

Write a program to convert Celsius to Fahrenheit

CODE :

```
C = 37.5
```

```
F = (C * 1.8) + 32
```

```
print("Temperature in Fahrenheit:", F)
```

OUTPUT :

Temperature in Fahrenheit:-99.5

CONCLUSION :

From the above experiment use conclude that the experiment has done successfully.

EXPERIMENT- 10

AIM OF THE EXPERIMENT :

Find the ASCII value of charades.

CODE :

```
A = 'Charades'  
print("The ASCII values are:", [ord(i) for i in A])
```

OUTPUT :

The ASCII value is [99, 104, 97, 114, 97,100, 101,115]

CONCLUSION :

From the above experiment we conclude that the experiment has done successfully.

EXPERIMENT- 11

AIM OF THE EXPERIMENT :

Write a program for simple calculator.

CODE :

```
first = int(input("Enter 1st number: "))
operator = input("Enter operator (+, -, x, %, /): ")
second = int(input("Enter 2nd number: "))
if operator == '+':
    print(first + second)
elif operator == '-':
    print(first - second)
elif operator == 'x':
    print(first * second)
elif operator == '%':
    print(first % second)
elif operator == '/':
    print(first / second)
else:
    print("Invalid operator")
```

OUTPUT :

```
Enter 1st no :- 6
Enter operator (+,-, X,%,/):+
Enter 2nd no:-2
Ans=8
```

CONCLUSION :

From the above experiment we conclude that the experiment has done successfully