



# **BACHELOR OF COMPUTER APPLICATIONS**

## **SEMESTER 5**

### **DCA3104**

### **PYTHON PROGRAMMING**

# Unit 1

## Introduction to Python Programming

### Table of Contents

SL No	Topic	Fig No / Table / Graph	SAQ / Activity	Page No
1	<a href="#">Introduction</a>	-	-	3 - 4
1.1	<a href="#">Learning Objectives</a>	-	-	
2	<a href="#">About Python Language</a>	-	<a href="#">1</a>	5 - 6
3	<a href="#">Need and Applications of Python Language</a>	<a href="#">1</a>	<a href="#">2</a>	7 - 9
4	<a href="#">Evaluation of Python Language</a>	-	<a href="#">3.I</a>	10 - 11
5	<a href="#">Python 2 vs. Python 3</a>	<a href="#">2</a>	<a href="#">4</a>	12 - 14
6	<a href="#">Features of Python Programming</a>	<a href="#">3</a>	<a href="#">5</a>	14 - 19
7	<a href="#">Popular Frameworks of Libraries</a>	-	<a href="#">6.II</a>	20 - 21
8	<a href="#">Summary</a>	-	-	22
9	<a href="#">Glossary</a>	<a href="#">4</a>	-	23 - 24
10	<a href="#">Case study</a>	-	-	25
11	<a href="#">Terminal Questions</a>	-	-	26
12	<a href="#">Answer Keys</a>	-	-	27 - 29
13	<a href="#">Suggested Books and E-References</a>	-	-	30

## 1. INTRODUCTION

The computer understands binary language. It is a language that has only two letters, one can say. These letters are 0 and 1. All other letters, words, numbers, and even sentences are formed from the combination of these two numbers. During the time when the first generation of computers were developed, scientists had to provide commands to the computer using this binary language. Though it can be learned and understood, it is highly complex and one cannot imagine performing larger, more complex functions using binary languages.

### STUDY NOTE

Lady Augusta Ada Lovelace is known as the first programmer of the world.

With the advancement of computer technologies in the coming generations, people moved on from Machine Language, that is, the language of 0s and 1s to Assembly Language. Assembly Languages are low level languages created as an intermediate between human language, primarily English and Machine Language. They use symbols and predefined keywords instead of 0s and 1s so that humans can easily provide commands to the computer in a comprehensive language. Abbreviations such as Add for addition, Sub for subtraction, etc., are used in low-level languages.

In the Assembly Language, an Assembler converts the commands provided by the programmer to Machine Language so that the computer can understand and execute them. Though, because of the addition of Assembler, the execution took much longer as compared to Machine Language.

After Assembly Language, high-level languages were developed. These are more similar to human languages and can be easily understood by anyone who does not have much knowledge or experience of programming languages. They have a set of grammar rules so that making instructions can be easier. These rules are called Syntax. It is different for different high-level languages. A compiler and interpreter are used to convert the high-level language into low-level or binary language. Various such high-level languages have been developed, primary ones being C, FORTRAN, COBOL, C++, Python, etc.

These days, high-level languages are created such that they can be easy to understand to the user. They should also have an easier syntax that is not restrained with too many rules. With the need for complex and large programs used for web-based applications, it is necessary that the programming language can handle numerous commands, libraries, and datatypes. It is also essential that the commands are executed quickly with least probability of presence of bugs and errors.

Python is such a high-level programming language that gives the freedom to the user while maintaining the capability to hold complex programs. It is open to customization and has one of the easiest syntaxes. Python was created with the aim to increase readability of code and allow the programmers to decrease effort and lines of code while writing larger programs.

## 1.1 Learning Objectives

*After studying the chapter, you will be able to:*

- ❖ *Understand the need of Python language and its various applications.*
- ❖ *Describe comparison of Python 2 and Python 3 along with evaluation of Python with different languages.*
- ❖ *Explain various features of Python language and programming.*
- ❖ *Understand the most popular frameworks of libraries in Python.*

## 2. ABOUT PYTHON LANGUAGE

The competitive times call not just for a high-level programming language, but also for a rapid application development (RAD) tool. Such tools should be able to design and support large applications seamlessly while providing features like fast response and a simple approach. All such needs are fulfilled by Python.

Python is one of the few languages that can be used as both, an interpreted and scripting language. It is a free and open-source language that is easy to learn and approach. Especially suited for Web applications, Python makes object-oriented programming simple and dynamic, enabling one to create large applications that are durable and adaptable.

Python was developed in 1991, with the aim that it could help reduce the time spent on monotonous and redundant tasks in programming. It is derived from the original scripting language ABC. At that time, ABC was used solely to teach programming by a few people.

Developed by Guido van Rossum, Python was originally created to perform and handle regular administrative tasks. It was employed at CWI in 1989, where Guido van Rossum was located at that time. It was released for the use of public in February 1991, after it was integrated into the Amoeba project at CWI.

### STUDY NOTE

Guido van Rossum created Python as a hobby project that can be a descendant of Unix.

Since then, the development of the language has been carried out at CNRI in Reston, Virginia, United States. By 2000, the developers of Python formed Pythonlabs under the BeOpen network. Pythonlabs has been maintained and organised by Guido van Rossum, among other primary developers of Python. Later in the year, Pythonlabs moved under Digital Creations. It has since been working to develop and enrich the language with more and better features. All the features, properties, etc., that are added to the language are managed by Python Software Foundation, a non-profit organisation.



**SELF-ASSESSMENT QUESTIONS - 1**

1. The developer of Python was \_\_\_\_\_.
2. Pythonlabs were first under \_\_\_\_\_ before moving under Digital Creations.
3. Python Software Foundation is a non-profit organisation. (True / False)
4. Python is an open-source and free low-level language. (True / False)
5. Python was released for the use of public in \_\_\_\_\_.



### 3. NEED AND APPLICATIONS OF PYTHON LANGUAGE

Python is the backbone of various web-based applications such as YouTube, Dropbox, etc. Its functionality and features such as supporting cross-platform operating systems make it a great tool for various applications. Here are the real-life based applications where Python is needed and highly used by top-notch companies.

#### Web Development



Source: Pinterest

**Fig 1:** Applications of Python

Python supports various frameworks. This increases the security, scalability, and speed at which a web-based application or website is developed and coded. There are multitudes of framework that support the integration of HTTPS, FTP, SSL, and much more. Frameworks like Flask, Django, and more can help the processing of JSON, E-Mail, XML, and other such protocols.

### **Machine Learning and Artificial Intelligence**

The future is in machine learning and artificial intelligence. It is imperative to learn and develop languages that can enhance these technologies and Python supports the experience. For machine learning, the computer should be able to come up with an algorithm based on its experience and learn by itself. This can be done by libraries such as NumPy, Pandas, and more which are already present in Python.

### **Data Science**

Understanding data and visualising it can help avoid risk and increase profits. The computer should be able to understand which data is relevant for a task, analyse it, and provide the results accordingly. Through libraries such as Pandas and NumPy, Seaborn, and Matplotlib, you can visualise and analyse the data. A python is an unmatched tool for data scientists.

#### **STUDY NOTE**

Companies like NASA, Google, Nokia, and IBM are based on Python.

### **Game Development**

Python also includes libraries that support 3D game engine such as PySoy, PyGame, etc. These libraries ease and enhance the process of building a game. Python also ensures that the game processes faster and without any bugs and errors.

### **Business Applications**

Business applications should be scalable to support the business as it grows. They should also be readable and extensible. Thus, languages like Python provide all such features to the developers. Various platforms such as Tryton can create applications that are coded in Python.



## Desktop GUI

Using the Tkinter library available in Python, you can create desktop applications as well. From simple applications such as post-it notes and calculators to high-end applications, toolkits such as wxWidgets, PYQT, and others provided in Python can help you create and develop various such applications.

## Web Scraping Applications

There are various applications such as job listings, research and development, comparison of prices, etc. when one needs to assess and analyse data from various websites and huge databases. Python's libraries such as BeautifulSoup can scrape and analyse such data without compromising speed and efficiency.

Along with these, Python is used to develop various other applications such as audio and video supporting apps, CAD applications, embedded applications, and much more. Its flexibility and scalability make it suitable for high-level computations and complex codes.

### SELF-ASSESSMENT QUESTIONS - 2

6. \_\_\_\_\_ and \_\_\_\_\_ libraries of Python are used in game development.
7. Through libraries such as Pandas, NumPy, etc., one can create programs for \_\_\_\_\_.
8. Python supports integration of HTTPS. (True / False)

## 4. EVALUATION OF PYTHON LANGUAGE

When comparing Python with other languages, the one benefit that makes Python unique is that it works as a bridge between interpreted and scripted languages. It holds the benefits of both types of languages, making it a much more accessible option than its other counterparts.

Most scripted languages are slower than interpreted languages. However, Python provides the user with the option to write scripts using Python without compromising on the speed of the compiler. Python is usually compared to C and C++ because of its similar syntax. It is majorly used to test applications written in C and C++. Python works better than these two high-level languages as it provides various benefits over them such as memory allocation and management, avoiding reference errors. The Python interpreter is responsible for memory management. Also, with the rich library available in Python, you can write much shorter and compact programs.

Another such language that can be compared by Python is Perl. Perl is another popular language that is used for data extraction and text manipulation. It is mostly used for system administration. However, Perl follows a difficult and much more complex syntax as compared to Python. Perl can be used to create powerful CGI (Common Gateway Interface) scripts but the code is difficult to comprehend and people working on a large project often find it hard to understand each other's code.

Tcl is also one of the programming languages that is often compared by Python. Tcl is as powerful as Python and is an easy-to-use scripting language. However, the data types for variables at Python are much richer than those available at Tcl. Tcl and Python have a similar toolkit for developing GUI applications, called Tk. Though, people tend to favor Python while developing such applications for its richer library.

Python is an object-oriented language that works similar to Java. However, Java requires larger codes that take longer to write, understand, and compile. It also does not offer features such as a rapid development environment and dynamic typing. But Python is slower than Java and is much less portable. To integrate the features of Python and classes available in Java, JPython was

### STUDY NOTE

Python can support multiple assignments in one statement.

developed. JPython provides various salient features such as providing a true object-oriented programming environment and a scripting environment for Java.

### Activity I

Research various famous websites and applications such as Snapchat, Instagram, Youtube, etc. For a list of apps and websites that are based on Python and its versions and those which are based on other programming languages.

### SELF-ASSESSMENT QUESTIONS - 3

9. \_\_\_\_\_ is an easy-to-use scripting language.
10. \_\_\_\_\_ is the amalgamation of features of Java and Python.
11. Python has a similar syntax as C and C++. (True / False)
12. In Python, memory management is done by \_\_\_\_\_.

## 5. PYTHON 2 VS. PYTHON 3

Python 2 was first developed and released in 2000. It has been one of the most successful versions of Python, the most popular one being Python 2.7. It was released in 2010 and is still in use by various companies.



**Source:** DEV Community

**Fig 2:** Python 2 vs Python 3

Python 3 was created in 2008. At that time, most programmers and companies were already using and were satisfied with the previous version of the language. However, the tools and features added to Python 3 soon spread amongst the community, leading to a debate among the programmers regarding which is the better version.

With the constant development and betterment of Python 3 and the strength of version Python 3.5, it was clear that the newer version of Python is the clear winner of the debate and should be the choice of new programmers who are just delving into the world of Python.

Now, most programmers agree that Python 2 dictates the legacy and the strength of the programming language whereas for the newer and faster web-based applications suited to the speed of the current generation, Python 3 and higher versions are much more suitable. Here is a comparison between both versions to aid your understanding.

## Incompatible Libraries

The libraries that were developed for Python 2 are incompatible with Python 3. Also, seeing that Python 3 is the future, the libraries developed are strictly compatible with this version only.

## Unicode Support

In Python 2, strings were stored in ASCII by default. If you want to store a string in Unicode, then you will have to specify a “u”. However, in Python 3, all the strings are stored in Unicode by default. Unicode is much more versatile as compared to ASCII. It can also save various symbols, numerals, emojis, Roman numerals, and much more.

## Better Integer Division

In Python 2, if you divide two integers that will result in a decimal answer, then the compiler will round off the answer to the nearest integer. Hence, to get an answer in decimal form, you will have to specify decimals in the subtraction. For example, use 2.0 instead of 2.

In Python 3, the compiler is expected to return the result in decimal if the answer is decimal. You do not have to add a decimal in the question.

Various companies such as Instagram and Facebook are moving from Python 2 to Python 3. This is because the newer function has a faster runtime. Also, there will be no developments in Python 2 after the version Python 2.7, making it a thing of the past.

### SELF-ASSESSMENT QUESTIONS - 4

13. \_\_\_\_\_ was the last version of Python 2, after which its development was stopped.
14. \_\_\_\_\_ and \_\_\_\_\_ are some of the companies that are upgrading from Python 2 to Python 3.
15. In Python 3, strings are stored in \_\_\_\_\_ by default.
16. You need to specify decimals to get the result in decimals in \_\_\_\_\_.



## 6. FEATURES OF PYTHON PROGRAMMING MEASUREMENT OF SOCIAL CLASS

The best feature of Python can be its ease. Programmers can pick Python as their first language to learn. You can easily develop large applications using it. It can also act as a bridge to connect the programmers to other, more advanced programming languages. Here are some of the prominent features of Python that make it a flexible and strong option for most developers.

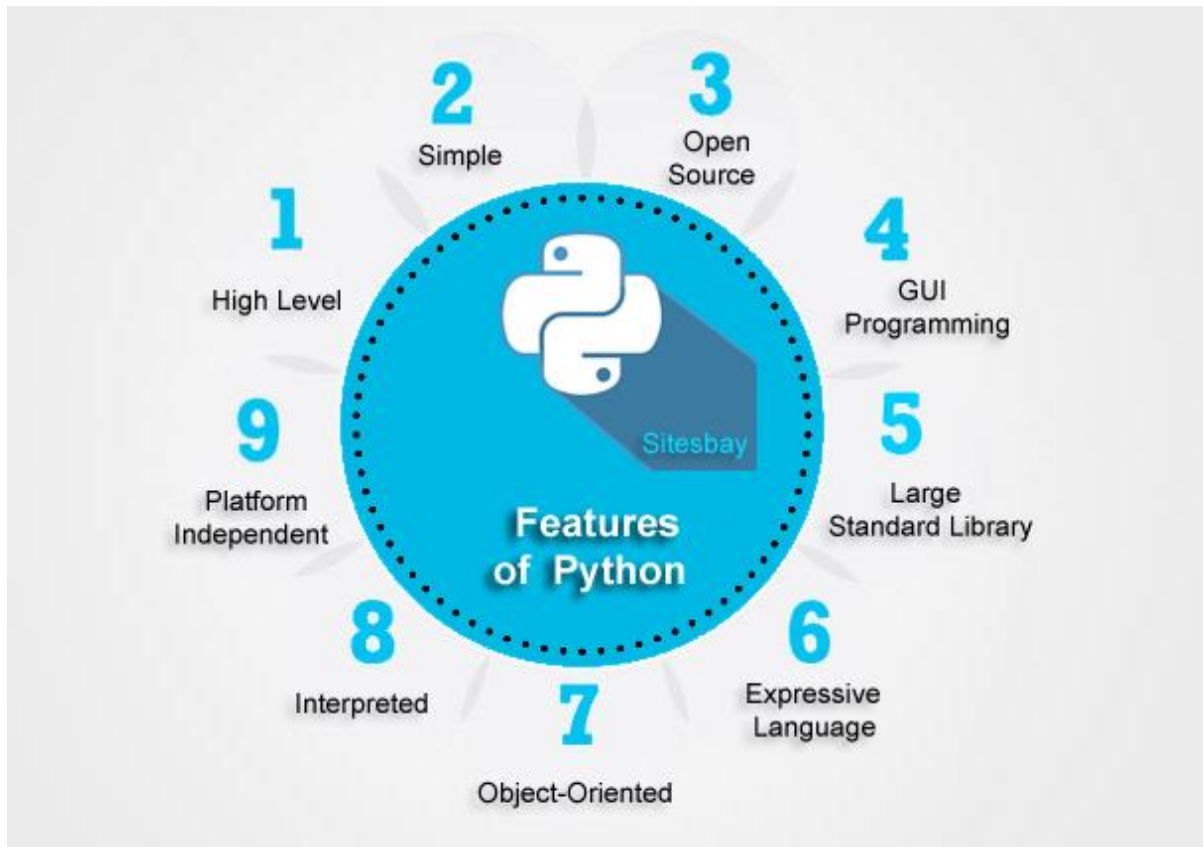
### Easy to Understand and Use

You can understand code written in Python even if you have no background in programming. It has an easy syntax that uses the common language without the presence of complex keywords. Lack of too many keywords and simple semantics used in Python ensures that one can learn it with relative ease. It is a simplified version of primary programming languages such as C, Algol, Pascal, etc., and follow the same syntax.

Though an object-oriented language, it is not necessary to use the concept in Python. One can run various applications and operations without it, a feature that makes it easier to approach than C++.

Python uses very few symbols within the code, to make coding simple and straightforward. It does not need a symbol to show the end of a command, nor does it need curly braces ({} ) to indicate a block of statements. One can use indentation to group statements as a single code block, ensuring lesser bugs and easier comprehension of the code.





*Source: Sitesbay*

**Fig 3: Features of Python Language**

### **High Level Language**

Python is an intuitive language. While using C and other such languages, one has to declare the datatype of the variable before storing any data in it. Also, other features such as system call, strings or varying lengths, etc., need a chunk of code to be written in C. Python reduces the time and energy spent on understanding and writing the code by providing built-in modules to its users. Lists, dictionaries, and other high level data types that can be used in Python are already built into Python. Thus, using them requires lesser time and understanding of the language. Also, Python assumes the datatype of a variable once you assign a value to it, decreasing the length of the code.

## Scalable

Scripting languages such as Unix work seamlessly and are easy to use while performing easy and straightforward tasks. However, once you start adding more lines to the code or features in the script, it becomes complicated and slow. These days, even smaller applications need large code and scripts to include all the required features. Hence, these scripting languages cannot keep up with today's need of fast and scalable programming languages. Through Python, one can merge one code to another or insert new features in an existing code without worrying about the speed of the code. Python ensures more functionality by allowing the user to change and modify the code by adding plug-ins and advancing architecture of an existing script.

## Object-Oriented Language

An Object-oriented language means that each component in a code written in Python is treated as an object. Though not necessary to use the concept of OOP (object-oriented programming) in Python, it is nonetheless a strong tool. You can create smaller programs without indulging or understanding the concept of OOP, making it easier to learn and understand the language. For larger programs, you can use object-oriented class hierarchies with every attribute in the class possessing a name. Python can accept any number of arguments.

## Interpreted Language

In conventional interpreted languages, the program is not interpreted in binary language. Thus, the process takes longer and is slower than other languages that support the compiler. Though, Python is an interpreted language where code is already byte-compiled, that is, available in binary language, when it is added to the interpreter. The feature enables Python to act as an intermediary between compiled and interpreted language, as it has the features of both.

### STUDYNOTE

As per a survey in 2015 conducted in UK, it was found that more people prefer learning Python than French.

Python is an interpreted language as the interpreter can print the result when in command-line mode. Signs such as ">>>" work as a prompt, indicating that the interpreter is ready to use.

Another way a program can be interpreted is through executing a file. Such a file is called a script and is saved with an extension of .py. To execute the file, the name of the script has to be written for the interpreter to understand.

### **Flexible**

A large code can be hard to understand and debug. Python allows fragmenting the code into smaller, more manageable parts. These parts, or modules, can still interact with each other and connect with the other in-built modules. Python allows flexibility to the user by letting them to interact with the external environment while using the module.

### **Library**

Python comes with a rich library of modules that can be integrated into the program. The library is updated and maintained by Python Standard Library. These modules can ease the process of coding and shorten the length of the code as well. The library can be used to write codes that perform the tasks of HTTP, FTP, POP, etc. You can write applications for various functions with the help of the in-built modules of python that are used to develop graphic user interface, downloading a webpage, and various others.

### **Memory Management**

Memory management is an essential part of programming when it comes to languages like C, C++, etc. Without proper memory management such as deleting the assigned memory once the task of the variable is completed, the programmer may face issues like memory leakage. This makes the program unnecessarily long. In Python, the task of memory management is handled by the interpreter. The interpreter is bound to make less or no errors and decrease the development time of the program as well.

## **Web Scripting Support**

Python is a dynamic language. It can not only handle complex programs with ease, but also supports several environments. Web applications based on the internet and intranet need such features in a programming language for them to work smoothly. Through the rich library system and flexibility of Python, you can write several complex programs and codes that support advanced features such as HTML, SGML parsing, XML, CGI scripts, etc.

## **Database**

Through the in-built modules available in the library of Python, you can handle several flat file databases. It has the capability to provide interfaces for major databases. One of the most essential and multi-featured databases of Python is Python API. It eases the process of writing codes for applications through which you can communicate with different databases.

## **Object Distribution**

Through Python, you can use objects that are distributed over various platforms. The code can communicate and interact with objects that are coded or available in other languages. One such example is passing data to COM components.

## **Embeddable**

Python interpreter can be extended with low-level modules. These extensions let the programmer customise the program according to the needs and interface of the application needed. You can connect Python to external libraries and modules to implement new data types in the code. Extension modules on Python can be programmed in other languages such as C and C++ for CPython and in Java for JPython.

## **GUI Programming**

Python holds a graphic user interface library Tkinter that allows the user to create GUI applications for different systems and libraries such as Windows MFC, X Window system of Unix, Macintosh, and more. Tkinter has the object-oriented interface of TK GUI API.



**Portable**

Python is written as ANSI C. Hence, it is not tied to one particular operating software and can be used over a large number of platforms without losing any of its features or credibility. One can write, execute, test, and upload a program written in Python in various environments including Macintosh, Linux, and Windows. However, the application will run as per the commands specified in it. If the commands are specified for one environment, for example, Linux, then the program will not be able to implement properly when interpreted in other environments.

**Freeware**

Python is an open-source language and thus can be redistributed freely. Anyone can use Python's source in any way they want without harming or putting it at risk. Also, the users are dissuaded from trying to take over the copyright of the source code. The programmers and users around the world can freely use Python to create codes using modules from the library in byte-compiled form.

**Exception Handling**

Through exception handling, Python can generate a stack trace of errors. This is when due to an error or bug in the program, Python exits. With the help of the trace, you can detect errors during run time without the need of statements that can check errors. The programmer does not have to spend a lot of time debugging the code as the Exception Handler can detect the problem, diffuse it, and perform a maintenance check.

**SELF-ASSESSMENT QUESTIONS - 5**

17. Python is considered a scripting language. (True / False)
18. Python does not need to be compiled before being run. (True / False)
19. It is necessary to understand and implement object-oriented programming while using Python even for the simplest programs. (True / False)
20. Python is mainly used for web-based applications. (True / False)

## 7. POPULAR FRAMEWORKS OF LIBRARIES

Python is majorly used to write various web-based applications. To make the process of coding easier, various modules are created and stored in the library of python. The collection of these modules is called a framework. Frameworks allow one to integrate codes for low-level details and help the programmer to save time.

By using various frameworks, one can automate the implementation of common commands, thus giving the user the time to spend on web-based application logic and algorithms.

Here are some of the most popularly used frameworks in Python.

### Django

It is an open-source and free framework. Based on the DRY principle, or Don't Repeat Yourself, Django uses ORM mappers. Object-relational mapper is used to manipulate the data from a database. It uses an object-oriented paradigm. Through Django, one can get various essential features in their code such as ORM, template engine, URL routing, authentication, and database schema migrations.

### Web2Py

It is another full-stack framework for Python that is scalable and open-source. It can be installed without any special prerequisites. Web2Py can be implemented and run-in various environments and different platforms. It protects the code against cross-site scripting, SQL injection, etc. It comes with its own code editor, debugger, and has a one-click deployment as well. It does not support Python 3 as of yet.

### Flask

This lightweight micro-framework is customisable and adapts to the needs of the user. It provides multitudes of features such as a quick and seamless debugger, built-in development server, secure cookie support. It also supports HTTP request handling, can plug in any ORM, supports Jinja2 templating, and is Unicode based.



## Bottle

The bottle provides default features such as templating, routing, a built-in development server, an abstraction layer over the WSGI standard. It is a micro-framework whose server can support all other WSGI-capable HTTP servers. It can also access data to create file uploads, cookies, headers, etc.

## CherryPy

CherryPy is especially targeted to ease the process of creating web-based applications. It is an open-source framework that makes coding for web-based applications similar to object-oriented programming. It packs in features such as caching, authentication, encoding, plug-in system flexibility, and the ability to run on different platforms and in different environments. It can run multiple HTTP servers simultaneously and offers built-in support for profiling, testing, and coverage.

Different frameworks are suited for various applications. All offer different functionalities that can be chosen as per the purpose of the program. You will have to assess the need for the program and choose the framework that provides the required features.

### Activity II

Create an algorithm for any program or task of your choice. Find the frameworks that can help you ease the algorithm and shorten your time if you were to code the algorithm using Python.

### SELF-ASSESSMENT QUESTIONS - 6

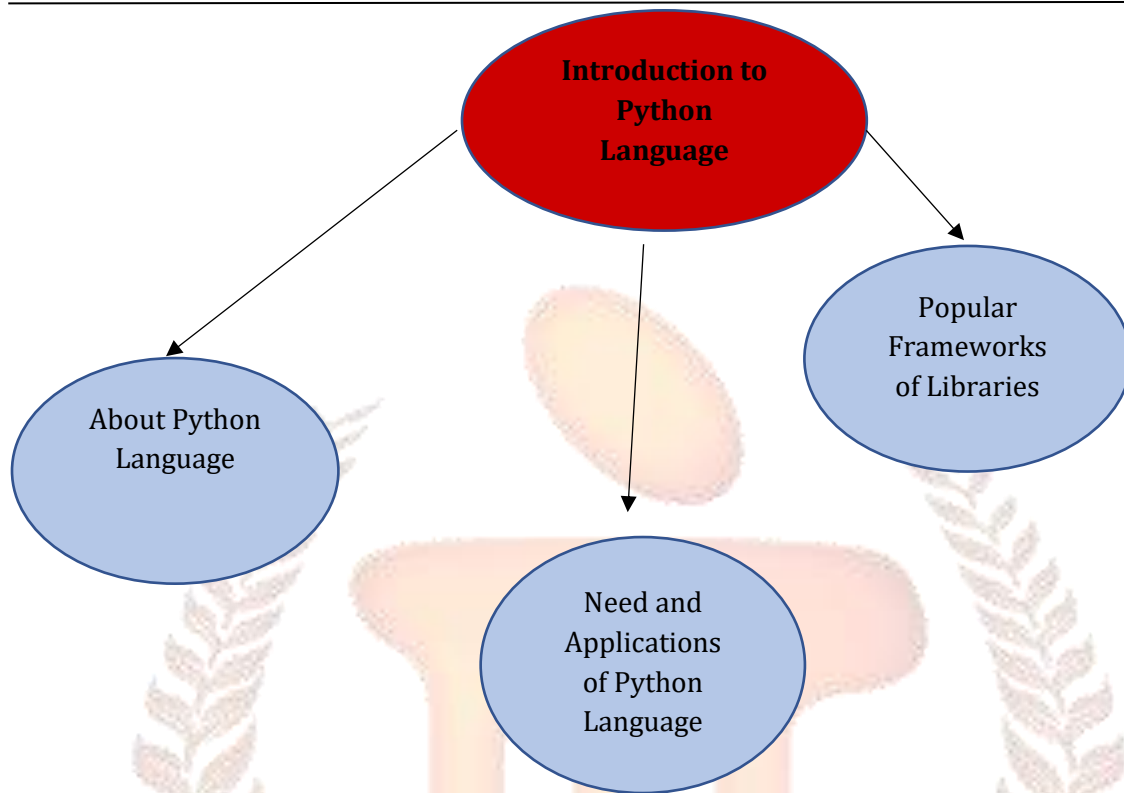
21. Flask is a \_\_\_\_\_ kind of framework.
22. \_\_\_\_\_ is the collection of modules that can be integrated in a Python code.
23. \_\_\_\_\_ can run various HTTP servers at the same time.
24. Web2Py can support both Python 2 and Python 3. (True / False)

## 8. SUMMARY

- Python is a free, open-source programming language. It is a scripted language with features of interpreted language as well.
- Python is used for various web-based applications. Along with that, it is used to develop business related applications, game development, machine learning, and data science.
- Python language's most successful version was Python 2. It was first developed in 2000.
- Most on the companies and programmers are now moving towards Python 3 after the development of Python 2 was stopped.
- Python was created to put emphasis on readability and scalability of a language.
- Python can handle larger programs, as opposed to other scripting languages such as Perl.
- There are various frameworks that can be integrated with Python. Some of these are full-stacked and some are micro-frameworks.
- Django, Flask, Bottle, etc., are some of the most famous frameworks for Python.

## 9. GLOSSARY

- **High-level language:** Programming languages that resemble human language the most. These come with compiler and interpreter to convert the commands into binary language.
- **Object-oriented programming:** OOP or object-oriented programming is a paradigm that includes the concept of classes and objects.
- **Scripted language:** Scripted language is one that is translated into binary language when the code is run.
- **Interpreted language:** Interpreted language is one in which code is compiled before it is run.
- **Freeware:** Freeware is software that is distributed without any cost. A freeware can be used by a programmer as per their desires.
- **Interpreter:** It is a computer program that executes the commands written in a scripted language without needing to compile them beforehand into machine language.
- **Keywords:** Reserved words in a programming language that are used to define a syntax are called keywords.
- **Frameworks:** It is a collection of modules or packages to help the developer write a code for web applications.



**Fig 4: Conceptual Map**

## 10. CASE STUDY

### Instagram runtime using Python

Instagram uses Python as a runtime use to execute their code. They have a database of more than 800 million active users. Python provides them with efficiency and speed needed to maintain such a database and execute the code seamlessly.

They collect and manage their data with an interpreter called InstaLab. It collects information of a user without affecting the user's experience on the app and website.

They are always looking for ways that can decrease the workload in the interpreter. According to the stats, 90% of the instructions handled by the interpreter are related to operand stack manipulation, control flow, and attribute access.

It was a complete success by using the python in the Instagram to track the runtime and to store all the needed information's.

*Source: Instagram Engineering*

### Questions

- 1) Find out more about the techniques used by Instagram to decrease their runtime and load on the interpreter.
- 2) Discuss various frameworks that applications like Instagram can use to decrease the lines of code.

## 11. TERMINAL QUESTIONS

### SHORT ANSWER QUESTIONS

- Q1. What is object-oriented programming?
- Q2. What do you understand about scripted languages?
- Q3. Why is Python considered an interpreted language?
- Q4. State any two uses of Python.
- Q5. What is the use of Unicode Support in Python?

### LONG ANSWER QUESTIONS

- Q1. Compare the features of Python 2 and Python 3.
- Q2. List any five features of Python. How do these features act as an advantage for the language?
- Q3. Describe the frameworks of libraries in Python.
- Q4. Give a detailed account about the applications of Python.



## 12. ANSWERS

### SELF ASSESSMENT QUESTIONS

1. Guido van Rossum
2. BeOpen Networks
3. True
4. False
5. February 1991
6. PySoy and PyGame
7. Machine learning
8. True
9. Perl
10. JPython
11. True
12. Interpreter
13. Python 2.7
14. Instagram and Facebook
15. Unicode
16. Python 2
17. False
18. True
19. False
20. True
21. Full-stack
22. Framework
23. CherryPy
24. False

## TERMINAL QUESTIONS

### SHORT ANSWER QUESTIONS

**Answer 1:** Object-oriented programming is one in which each data and functions are attributed as objects. An abstract data type is created by the developer. These objects are defined as classes. The objects are used to provide structure to the programming.

**Answer 2:** Scripted or scripting languages are those programming languages where instructions are written for run-time environment. They do not need compilation. It is usually used for integrating with other programming languages and environments.

**Answer 3:** Python is considered an interpreted language because the code goes through an interpreter while being run. The interpreter turns the keywords and other commands to machine language so that the computer can process it.

**Answer 4:** Python is used to develop web-based applications that support protocols such as HTTPS and FTP. It is also used to create seamless and faster 3D game through various libraries present in Python.

**Answer 5:** In Python 2, strings were stored in ASCII by default. If you want to store a string in Unicode, then you will have to specify a “u”. However, in Python 3, all the strings are stored in Unicode by default. Unicode is much more versatile as compared to ASCII. It can also save various symbols, numerals, emojis, Roman numerals, and much more.

### LONG ANSWER QUESTIONS

**Answer 1:** Python 2 stores strings in ASCII code by default whereas they are stored in Unicode in Python 3. No small brackets are needed for keywords in Python 3 as opposed to Python 2. Python 2 rounds off the answer of division if it is in decimal form. Python 3 gives a decimal answer. The development of Python 2 has stopped; hence no run time developments will be done in that version.

**Answer 2:** Python is an interpreted language. That means that, unlike languages like C and its variants, Python does not need to be compiled before it is run.

Python is suitable for object orientated programming in that it allows the definition of classes along with composition and inheritance.

Python can be used in many applications – web applications, automation, scientific modelling, big data applications and many more.

**Answer 3:** Frameworks are collection of modules that decrease the number of lines of commands. In Python, various frameworks can be integrated. These are divided into two subcategories, full stacked and micro frameworks. The following are the most prominent ones used by developers.

Django

Flask

Bottle

Web2Py

CherryPy

**Answer 4:** Python is mostly used to develop web-based applications. It is also used for the following purposes.

**Game development:** Using PyGame and other libraries, 3D games can be developed using Python.

**Data Science:** Python can manage a huge database and analyse it. Thus, it can be used by data scientists to obtain results.

**Web Scraping Applications:** Python can analyse various websites and compare the data in them.

Other applications include business-based applications, desktop GUI, etc.

### 13. SUGGESTED BOOKS AND E-REFERENCES

#### BOOKS:

- Eric Matthes (2016), Python Crash Course: A Hands-On, Project-Based Introduction to Programming.
- John M. Zelle (2009), Python Programming: An Introduction to Computer Science (Preliminary Second Edition).
- Mark Lutz (2011), Python Programming: A Powerful Object-Oriented Programming (Fourth Edition).
- Sebastian Raschka (2017), Python Machine Learning - Machine Learning and Deep Learning with Python (Edition 2)

#### E- REFERENCES:

- Python Programming Certification Training Course, last viewed on March 23, 2021 < <https://www.edureka.co/python-programming-certification-training> >
- Python Tutorials and Sample Programs, last viewed on March 23, 2021 < <https://www.w3schools.com/python/> >
- History of Python, last viewed on March 23, 2021 < [https://en.wikipedia.org/wiki/History\\_of\\_Python](https://en.wikipedia.org/wiki/History_of_Python) >