PYTHON PROGRAMMING

CSE3011

Course Objectives

This course will introduce the Python Programming language, its functionality, code constructs, and its applications. This course is devised for following objectives,

1. To study object oriented paradigm in Python.
2. To develop their skill set using Python.
3. To familiarize with the functionalities and applications of Python.

Course Outcomes

- Students will be able to solve problems, explore real-world software development challenges, and create practical and contemporary applications. At the completion of this course, students should be able to do the following:

1. Understand and use the Object Oriented paradigm in Python
2. Use the IO model in Python to read and write disk files.
3. Write Python programs using collections, regular expression, classifying and categorizing text.

UNIT-1

A Brief History of Python, Different Versions, Python 2 vs Python 3, Installing Python, Environment Variables, Executing Python from the Command Line, Editing Python Files, Basic Python Syntax, String Values, String Operators, Numeric Data Types Conversions, Simple Input and Output, Language components - Control Flow structures and Syntax - Relational Operators - Logical Operators - Bit Wise Operators, Python for Windows

UNIT-2

Conditions, boolean logic, logical operators, ranges, Control statements: if-else, loops (for, while), Flow control, Functions, Scoping, Exceptions, Input and output, Modules, Collections, Lists, Tuples, Sets, Dictionaries, Modules, Standard Modules, Regular Expressions, Quantifiers, Basic String Operations

UNIT-3

Principles of Object Orientation, Classes in Python, Creating Classes, Instance Methods, Access Specification, data modeling, persistent storage of objects, inheritance, polymorphism, operator overloading, abstract classes, exception handling, try block

UNIT-4

- File Handling, Writing Data to a File, Reading Data From a File - Additional File Methods: Using Pipes as Data Streams, Handling, IO Exceptions, Working with Directories, Metadata, File Organization, Database Programming - Generic Database Connectivity using ODBC, Postgres connection in Python, MySQL connection in Python.

UNIT-5

Graphical user interfaces, event-driven programming paradigm, tkinter module, creating simple GUI, buttons, labels, entry fields, dialogs, widget attributes - sizes, fonts, colors layouts, nested frames, Multithreading, Networks, and Client/Server Programming, introduction to HTML, interacting with remote HTML server, running html-based queries, downloading pages;CGI programming, programming a simple CGI form.

History of the Python

- Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI) in the Netherlands as a successor of a language called ABC.
- Guido is Python's principal author, although it includes many contributions from others. The last version released from CWI was Python 1.2.
- In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI) in Reston, Virginia where he released several versions of the software.
- Python 1.6 was the last of the versions released by CNRI.
- In 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team.
- Python 2.0 was the first and only release from BeOpen.com.

- Following the release of Python 1.6, and after Guido van Rossum left CNRI to work with commercial software developers, it became clear that the ability to use Python with software available under the GNU Public License (GPL) was very desirable.
- CNRI and the Free Software Foundation (FSF) interacted to develop enabling wording changes to the Python license.
- Python 1.6.1 is essentially the same as Python 1.6, with a few minor bug fixes, and with a different license that enables later versions to be GPL-compatible.
- Python 2.0.1 is a derivative work of Python 1.6.1, as well as of Python 2.0.

- After Python 2.0 was released by BeOpen.com, Guido van Rossum and the other PythonLabs developers joined Digital Creations.
- All intellectual property added from this point on, including Python 2.0.1 and its alpha and beta releases, is owned by the Python Software Foundation (PSF), a non-profit modeled after the Apache Software Foundation.
- 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.
- The following programming languages influence Python:
  - ABC language.
  - Modula-3

What is Python?

- Python is a popular programming language. It is used for:
  1. web development (server-side),
  2. software development,
  3. mathematics,
  4. system scripting.
- What can Python do?
  - Python can be used on a server to create web applications.
  - Python can be used alongside software to create workflows.
  - Python can connect to database systems. It can also read and modify files.
  - Python can be used to handle big data and perform complex mathematics.
  - Python can be used for rapid prototyping, or for production-ready software development.

Python Version List

Python programming language is being updated regularly with new features and supports. There are lots of update in Python versions, started from 1994 to current release.

Python Version	Released Date
Python 1.0	January 1994
Python 1.5	December 31, 1997
Python 1.6	September 5, 2000
Python 2.0	October 16, 2000
Python 2.1	April 17, 2001
Python 2.2	December 21, 2001
Python 2.3	July 29, 2003

Python Version	Released Date
Python 2.4	November 30, 2004
Python 2.5	September 19, 2006
Python 2.6	October 1, 2008
Python 2.7	July 3, 2010
Python 3.0	December 3, 2008
Python 3.1	June 27, 2009
Python 3.2	February 20, 2011
Python 3.3	September 29, 2012
Python 3.4	March 16, 2014
Python 3.5	September 13, 2015
Python 3.6	December 23, 2016
Python 3.7	June 27, 2018
Python 3.8	October 14, 2019

What is Python 2?

- Python 2 made code development process easier than earlier versions. It implemented technical details of Python Enhancement Proposal (PEP). Python 2.7 (last version in 2.x) is no longer under development and in 2020 will be discontinued.

What is Python 3?

- On December 2008, Python released version 3.0.
- This version was mainly released to fix problems which exist in Python 2.
- The nature of these change is such that Python 3 was incompatible with Python 2. It is **backward incompatible**
- Some features of Python 3 have been backported to Python 2.x versions to make the migration process easy in Python 3.
- As a result, for any organization who was using Python 2.x version, migrating their project to 3.x needed lots of changes.
- These changes not only relate to projects and applications but also all the libraries that form part of the Python ecosystem.

COMPARISON	PYTHON 3	PYTHON 2
Release Date	**2008**	**2000**
Function print	print ("hello")	print "hello"
Division of Integers	Whenever two integers are divided, you get a float value	When two integers are divided, you always provide integer value.
Unicode	In Python 3, default storing of strings is Unicode.	To store Unicode string value, you require to define them with "u".
Syntax	The syntax is simpler and easily understandable.	The syntax of Python 2 was comparatively difficult to understand.
Rules of ordering Comparisons	In this version, Rules of ordering comparisons have been simplified.	Rules of ordering comparison are very complex.
Iteration	The new Range() function introduced to perform iterations.	In Python 2, the xrange() is used for iterations.
Exceptions	It should be enclosed in parenthesis.	It should be enclosed in notations.
Leak of variables	The value of variables never changes.	The value of the global variable will change while using it inside for-loop.
Backward compatibility	Not difficult to port python 2 to python 3 but it is never reliable.	Python version 3 is not backwardly compatible with Python 2.
Library	Many recent developers are creating libraries which you can only use with Python 3.	Many older libraries created for Python 2 is not forward-compatible.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or a functional way.
- Python can be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Python Install

- Many PCs and Macs will have python already installed.
- To check if you have python installed on a Windows PC, search in the start bar for Python or run the following on the Command Line (cmd.exe):

  C:\Users*Your Name*>python --version

- To check if you have python installed on a Linux or Mac, then on linux open the command line or on Mac open the Terminal and type:

  python --version

- If you find that you do not have python installed on your computer, then you can download it for free from the following website: https://www.python.org/downloads/

Environment Variables

- If you've installed Python in Windows using the default installation options, the path to the Python executable wasn't added to the Windows **Path variable**.
- The Path variable lists the directories that will be searched for executables when you type a command in the command prompt.
- By adding the path to the Python executable, you will be able to access **python.exe** by typing the **python** keyword (you won't need to specify the full path to the program).
- https://www.javatpoint.com/how-to-install-python

Python Quickstart

- Python is an interpreted programming language, this means that as a developer you write Python (.py) files in a text editor and then put those files into the python interpreter to be executed.
- The way to run a python file is like this on the command line:

C:\Users*Your Name*>python helloworld.py

Where "helloworld.py" is the name of your python file.

- Let's write our first Python file, called helloworld.py, which can be done in any text editor.

helloworld.py

```
print("Hello, World!")
```

- Save your file.
- Open your command line, navigate to the directory where you saved your file, and run:

  C:\Users*Your Name*>python helloworld.py

- The output should read:

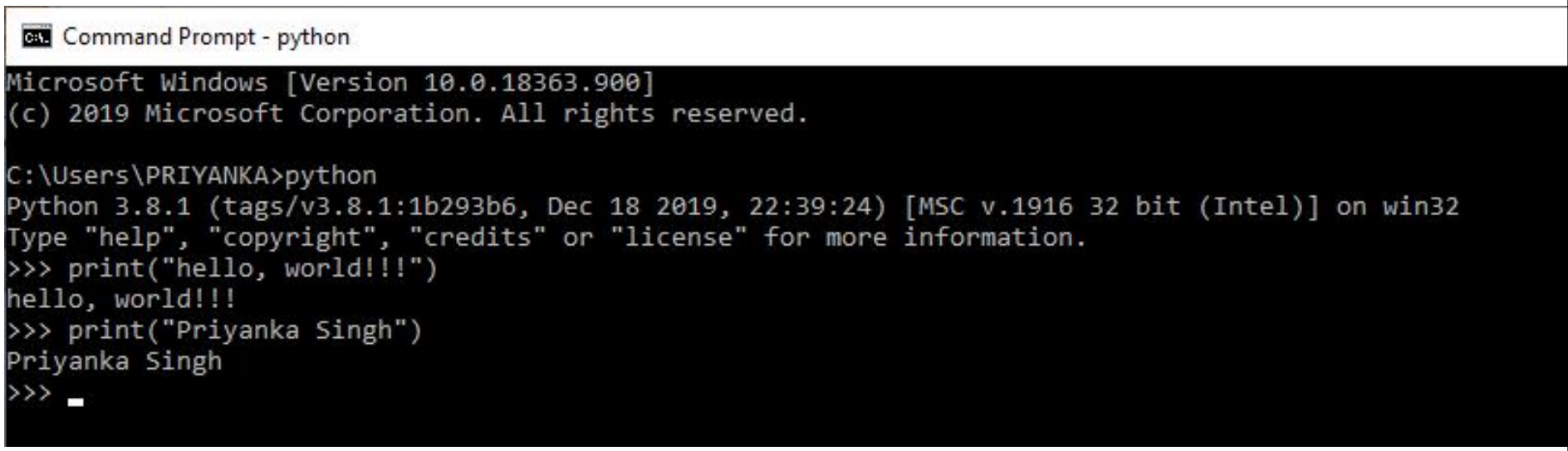
  Hello, World!

Executing Python from the Command Line

- o test a short amount of code in python sometimes it is quickest and easiest not to write the code in a file. This is made possible because Python can be run as a command line itself.
- Type the following on the Windows, Mac or Linux command line:

  C:\Users*Your Name*>python
- Or, if the "python" command did not work, you can try "py":

  C:\Users*Your Name*>py
- From there you can write any python, including our hello world example.



- Whenever you are done in the python command line, you can simply type the following to quit the python command line interface:

exit()

PyCharm Installation

- On Windows, we have an alternative like notepad or notepad++ to edit the code. However, these editors are not used as IDE for python since they are unable to show the syntax related suggestions.
- JetBrains provides the most popular and a widely used cross-platform IDE **PyCharm** to run the python programs.
- Install PyCharm on Windows operating system from https://www.jetbrains.com/pycharm/download/download-thanks.html?platform=windows