Open Elective – II: PYTHON PROGRAMMING (CS702OE)
COURSE PLANNER

I. COURSE OVERVIEW:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Python Programming is intended for software engineers, systems analysts, program managers and user support personnel who wish to learn the Python programming language. This Python for beginners training course leads the students from the basics of writing and running Python scripts to more advanced features such as file operations, regular expressions, working with binary data, and using the extensive functionality of Python modules. Extra emphasis is placed on features unique to Python, such as tuples, array slices, and output formatting.

This course PYTHON PROGRAMMING is an essential part of any Computer-Science education. To master the fundamentals of writing Python scripts, learn core Python scripting elements such as variables and flow control structures, discover how to work with lists and sequence data, write Python functions to facilitate code reuse ,use Python to read and write files, make their code robust by handling errors and exceptions properly, work with the Python standard library, explore Python's object-oriented features , search text using regular expressions and finally working with GUI (Graphical User Interfaces)

II. PRE-REQUISITES:

- A course on “Programming for Problem Solving using C”.

III. COURSE OBJECTIVIES:

1.Learn Syntax and Semantics and create Functions in Python.
2.Handle Strings and Files in Python.
3.Understand Lists, Dictionaries and Regular expressions in Python.
4. Implement Object Oriented Programming concepts in Python.
5. Build Web Services and introduction to Network and Database Programming in Python.

IV.COURSE LEARNING COUCOMES (CLOs):

CLO Code	CLO's	At the end of the course, the student will have the ability to:	Bloom's Taxonomy Levels	POs / PSOs mapped
CS311PC.01	CLO1	Examine Python syntax and semantics and be fluent in the use of Python flow control and functions	Level 4: Analyzing	PO1, PO2, PO3, PSO1, PSO2
CS311PC.02	CLO2	Demonstrate proficiency in handling Strings and File Systems	Level 2 : Understanding	PO1, PO2, PO3, PO4, PSO1, PSO2
CS311PC.03	CLO3	Create, run and manipulate Python Programs using core data structures like Lists, Dictionaries and use Regular Expressions.	Level 6 : Creating	PO1, PO2, PO3, PO4, PSO1, PSO2
CS311PC.04	CLO4	Interpret the concepts of Object-Oriented Programming as used in Python.	Level 2 : Understanding	PO1, PO2, PO3, PO4, PO5, PSO1, PSO2
CS311PC.05	CLO5	Implement exemplary applications related to Network Programming, Web Services and Databases in Python.	Level 6 : Creating	PO1, PO2, PO3, PO4, PO5, PSO1, PSO2

V. HOW PROGRAM OUTCOMES ARE ASSESSED:

Program Outcomes (POs)		Level	Proficiency assessed by
PO1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineeringfundamentals, and an engineering specialization to the solution of complex engineering problems.	2	Presentation on real-world problems
PO2	**Problem analysis**: Identify, formulate, review research literature, and analyze complexengineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	2	Assignments
PO3	**Design/development of solutions**: Design solutions for complex engineering problems anddesign system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.	3	Assignments
PO4	**Conduct investigations of complex problems**: Use research-based knowledge and researchmethods including design of	3	Mini/Major Projects

Program Outcomes (POs)		Level	Proficiency assessed by
	experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.		
PO5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modernengineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.	2	Mini/Major Projects
PO6	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assesssocietal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.	-	--
PO7	**Environment and sustainability**: Understand the impact of the professional engineering solutionsin societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.	-	--
PO8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms ofthe engineering practice.	-	--
PO9	**Individual and team work**: Function effectively as an individual, and as a member or leader indiverse teams, and in multidisciplinary settings.	-	--
PO10	**Communication**: Communicate effectively on complex engineering activities with the engineeringcommunity and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.	-	-
PO11	**Project management and finance**: Demonstrate knowledge and understanding of theengineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.	**2**	--
PO12	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage inindependent and life-long learning in the broadest context of technological change.	-	-

VI.HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED:

Program Specific Outcomes (PSOs)		Level	Proficiency assessed by
PSO1	**Foundation of mathematical concepts:** To use mathematical methodologies to crack problem using suitable mathematical analysis, data structure and suitable algorithm.	2	Assignments
PSO2	**Foundation of Computer System:** The ability to interpret the fundamental concepts and methodology of computer systems. Students can understand the functionality of hardware and software aspects of computer systems.	2	Mini/Major Projects
PSO3	**Foundations of Software development:** The ability to grasp the software development lifecycle and methodologies of software	3	Methodologies

Program Specific Outcomes (PSOs)		Level	Proficiency assessed by
	systems. Possess competent skills and knowledge of software design process. Familiarity and practical proficiency with a broad area of programming concepts and provide new ideas and innovations towards research.		

VII. MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course Learning Outcomes	Program Outcomes (PO)												Program Specific Outcomes (PSO)		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CLO1	3	2	2	3	2	-	-	-	-	-	-	2	2	2	1
CLO2	3	2	2	2	2	-	-	-	-	-	-	1	1	2	1
CLO3	3	3	3	3	2	-	-	-	-	-	-	1	1	2	1
CLO4	2	2	3	2	3	-	-	-	-	-	-	2	1	2	2
CLO5	1	2	2	3	2	-	-	-	-	-	-	1	2	2	2
AVG	2	2	2	3	2	-	-	-	-	-	-	1	1	2	1

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) - : None

VIII. SYLLABUS:

UNIT-I

Python Basics, Objects- Python Objects, Standard Types, Other Built-in Types, Internal Types, Standard Type Operators, Standard Type Built-in Functions, Categorizing the Standard Types, Unsupported Types Numbers - Introduction to Numbers, Integers, Floating Point Real Numbers, Complex Numbers, Operators, Built-in Functions, Related Modules Sequences - Strings, Lists, and Tuples, Mapping and Set Types

UNIT-II

FILES: File Objects, File Built-in Function [open()], File Built-in Methods, File Built-in Attributes, Standard Files, Command-line Arguments, File System, File Execution, Persistent Storage Modules, Related Modules Exceptions: Exceptions in Python, Detecting and Handling Exceptions, Context Management, *Exceptions as Strings, Raising Exceptions, Assertions, Standard Exceptions, *Creating Exceptions, Why Exceptions (Now)?, Why Exceptions at All?, Exceptions and the sys Module, Related Modules: Modules and Files, Namespaces, Importing Modules, Importing Module Attributes, Module Built-in Functions, Packages, Other Features of Modules

UNIT-III

Regular Expressions: Introduction, Special Symbols and Characters, Res and Python Multithreaded Programming: Introduction, Threads and Processes, Python, Threads, and the Global Interpreter Lock, Thread Module, Threading Module, Related Modules

UNIT-IV

GUI Programming: Introduction, Tkinter and Python Programming, Brief Tour of Other GUIs, Related Modules and Other GUIs WEB Programming: Introduction, Wed Surfing with Python, Creating Simple Web Clients, Advanced Web Clients, CGI-Helping Servers Process Client Data, Building CGI Application Advanced CGI, Web (HTTP) Servers

UNIT-V

Database Programming: Introduction, Python Database Application Programmer's Interface (DB-API), Object Relational Managers (ORMs), Related Modules

TEXT BOOK:

1. Core Python Programming, Wesley J. Chun, Second Edition, Pearson.

REFERENCE BOOK:

1.Think Python, Allen Downey, Green Tea Press
2. Introduction to Python, Kenneth A. Lambert, Cengage
3. Python Programming: A Modern Approach, VamsiKurama, Pearson
4. Learning Python, Mark Lutz, O'Really.

NPTEL RESOURCES:

1. NOC: Machine Learning, ML (Video):

   https://nptel.ac.in/courses/106106202/

1. NOC: Introduction to Machine Learning (Video):

   https://nptel.ac.in/courses/106105152/

2. NOC: Introduction to Machine Learning (Course sponsored by Aricent) (Video):

   https://nptel.ac.in/courses/106106139/

GATE SYLLABUS: NOT APPLICABLE

IES SYLLABUS: NOT APPLICABLE

XI. LESSON PLAN

Lecture No.	Unit No.	Topics to be covered	Content to be covered under each topic	Link for PDF	Link for Small Projects/ Numericals(if any)	Course Learning Outcomes	*Teaching Methodology	References
1	1	Introduction to python programming	Introduction, Download and Installation of python	https://drive.google.com/drive/folders/1acFXe12dKEJIvQNBpfEiWPw92tWpp9LN	NIL	CO1, CO2 & CO7	PPT, Pen, White Board	T1
2	1	Python Basics	● Statements and Syntax ● Variable Assignment ● Identifiers and Keywords ● Basic Style Guidelines ● Memory Management ● First Python Program		NIL	CO1, CO2 & CO7	PPT, Pen, White Board	T1
3	1	Objects-Python Objects,	Object Attribute Class Objects Instance Objects Method Objects		NIL	CO1, CO2 & CO7	PPT, Pen, White Board	T1
4	1	Standard Types	String List Tuple Dictionar		NIL	CO1, CO2 & CO7	PPT, Pen, White Board	T1
5	1	Other Built-in Types, Internal Types	Type ● Null object (None) ● File ● Set/Frozenset ● Function/Method ● Module ● ClasS Code ● Frame ● Traceback ● Slice ● Ellipsis ● Xrange		NIL	CO1, CO2 & CO7	PPT, Pen, White Board	T1
6	1	Standard Type Operators, Standard	Object Value Comparison Object Identity		NIL	CO1, CO2 & CO7	PPT, Pen, White Board	T1

		Type Built-in Functions	Comparison Boolean type() cmp() str() and repr() (and `` Operator) type() and isinstance() Python Type Operator and BIF Summary					
7	1	Categorizing the Standard Types, Unsupported Types	Storage Model Update Model Access Model char or byte pointer int versus short versus long float versus double		NIL	CO1, CO2 & CO7	PPT, Pen, White Board	T1
8	1	Numbers - Introduction to Numbers, Integers	. How to Create and Assign Numbers (Number Objects) How to Update Numbers How to Remove Numbers Boolean Standard (Regular or Plain) Integers Long Integers Unification of Integers and Long Integers		NIL	CO1, CO2 & CO7	PPT, Pen, White Board	T1
9	1	Floating Point Real Numbers, Complex Numbers	. Double Precision Floating Point Numbers. Complex Number Built-in Attributes		NIL	CO1, CO2 & CO7	PPT, Pen, White Board	T1
10	1	Operators, Built-in Functions, Related Modules	. Mixed-Mode Operations Standard Type Operators Numeric Type (Arithmetic) Operators Bit					

			Operators (Integer-Only)					
11	1	Sequences - Strings	· creation of string in Python Accessing string Python String Operations					
12	1	Lists, and Tuples	· creation and accessing of list in Python Operations on list creation and accessing of tuple Operations on tuple		NIL	CO1, CO2 & CO7	PPT, Pen, White Board	T1
13	2	Mapping and Set Types	· creation and accessing of set Operations on set					
14	2	FILES: File Objects	· Python File Handling Python Read Files Python Write/Create Files Python Delete Files	https://drive.google.com/drive/folders/1acFXe12dKEJIvQNBpfEiWPw92tWpp9LN				
15	2	File Built-in Function [open()], File Built-in Methods	· Create, Open and reading a File	https://drive.google.com/drive/folders/1acFXe12dKEJIvQNBpfEiWPw92tWpp9LN	NIL	CO1, CO2 & CO7	PPT, Pen, White Board	T1
16	2	File Built-in Attributes, Standard Files	· Attributes inside built function of Create, Open and reading a File		NIL	CO1, CO2 & CO7	PPT, Pen, White Board	
17	2	Command-line Arguments, File System,	· Write to an Existing File Passing		NIL	CO1, CO2 & CO7	PPT, Pen, White Board	

		File Execution, Persistent Storage Modules	Command-line Arguments pickle and marshal Modules DBM-style Modules shelve Module					
18	2	Related Modules Exceptions: Exceptions in Python, Detecting and Handling Exceptions	. What Are Exceptions? Exceptions in Python Detecting and Handling Exceptions Raising Exceptions Standard Exceptions Creating Exceptions		NIL	CO1, CO2 & CO7	PPT, Pen, White Board	T1
19	2	Context Management	. With Statement Context Management Protocol		NIL	CO1, CO2 & CO7	PPT, Pen, White Board	
20	2	*Exceptions as Strings, Raising Exceptions	. Exceptions as Strings Raise statement Using Raise statement		NIL	CO1, CO2 & CO7	PPT, Pen, White Board	
21	2	Assertions, Standard Exceptions	. Assert statement Python Built-In Exceptions		NIL	CO1, CO2 & CO7	PPT, Pen, White Board	T1
22	2	*Creating Exceptions, Why Exceptions? Why Exceptions at All?	. Creating Exceptions Need of exception		NIL	CO1, CO2 & CO7	PPT, Pen, White Board	T1
23	2	Exceptions and the sys Module	. Exceptions and the related Module		NIL	CO1, CO2 & CO7	PPT, Pen, White Board	T1
24	2	Related Modules:	. Exception-		NIL	CO1, CO3, CO4	PPT, Pen,	T1

		Modules and Files, Namespaces	Related Standard Library Modules			& CO7	White Board	
25	2	Importing Modules, Importing Module Attributes	. How to import module The import Statement Attribute of import module					
26	2	Module Built-in Functions, Packages, Other Features of Modules	. Module Built-in Functions, __import__ () globals() and locals() reload() Packages Using from-import with Packages Features of Modules Auto-Loaded Modules					
27	3	Doubt And revision		-	NIL	CO1, CO3, CO4 & CO7	PPT, Pen, White Board	T1
28	3	Regular Expressions: Introduction	. Motivation	https://drive.google.com/drive/folders/1acFXe12dKEJIvQNBpfEiWPw92tWpp9LN	NIL	CO1, CO3, CO4 & CO7	PPT, Pen, White Board	T1
	3	Special Symbols and Characters, Res and Python Multithreaded Programming: Introduction	. Special Symbols and Characters Common Regular Expression Symbols and Special Characters Matching More Than One RE Pattern with Alternation Re Module: Core Functions and Methods Common Regular Expression Functions and		NIL	CO1, CO3, CO4 & CO7	PPT, Pen, White Board	T1

			Methods Introduction of Multithreaded Programming					
29	3	Threads and Processes	. What Are Processes What Are Thread	https://drive.google.com/drive/folders/1acFXe12dKEJIvQNBpfEiWPw92tWpp9LN	NIL	CO1, CO3, CO4 & CO7	PPT, Pen, White Board	T1
30	3	Threads and the Global Interpreter Lock	. Global Interpreter Lock (GIL) Use of Global Interpreter Lock (GIL)		NIL	CO1, CO3, CO4 & CO7	PPT, Pen, White Board	T1
31	3	Thread Module	. Python Threading Modules Thread Module and Lock Objects		NIL	CO1, CO3, CO4 & CO7	PPT, Pen, White Board	T1
32	3	Threading Module, Related Modules	. Threading Module Objects Thread Class Thread Object Methods Create and using Thread Instance, Passing in Function Other Threading Module Functions		NIL	CO1, CO3, CO4 & CO7	PPT, Pen, White Board	T1
33	4	GUI Programming: Introduction,	Explain GUI Programming: Introduction What Are Tcl, Tk, and Tkinter	https://drive.google.com/drive/folders/1acFXe12dKEJIvQNBpfEiWPw92tWpp9LN	NIL	CO1, CO3, CO4 & CO7	PPT, Pen, White Board	T1
34	4	Tkinter and Python Programming	Understand Tkinter and Python Programming Tkinter Module: Adding Tk to your Applications Tk Widgets		NIL	CO1, CO3, CO5 & CO7	PPT, Pen, White Board	T1

35	4	Brief Tour of Other GUIs	Other GUIs		NIL	CO1, CO3, CO5 & CO7	PPT, Pen, White Board	T1
36	4	Related Modules and Other GUIs	GUI Systems Available for Python Define Brief Tour of Other GUIs		NIL	CO1, CO3, CO5 & CO7	PPT, Pen, White Board	T1
37	4	WEB Programming: Introduction	Understand Related Modules and Other GUIs WEB Programming: Introduction		NIL	CO1, CO3, CO5 & CO7	PPT, Pen, White Board	T1
38	4	WEB Programming: Web Surfing with Python	Describe WEB Programming Client/Server Computing		NIL	CO1, CO3, CO5 & CO7	PPT, Pen, White Board	T1
39	4	Creating Simple Web Clients Advanced Web Clients	. Creating Simple Web Clients Web Address Components Network Location Components Urlparse Module Advanced Web Client: a Web Crawler urllib2Module		NIL	CO1, CO3, CO5 & CO7	PPT, Pen, White Board	T1
40	4	CGI-Helping Servers Process Client Data	. Introduction to CGI CGI module		NIL	CO1, CO3, CO5 & CO7	PPT, Pen, White Board	T1
41	4	Building CGI Application Advanced CGI, Web (HTTP) Servers	. Building CGI Applications Setting Up a Web Server		NIL	CO1, CO3, CO5 & CO7	PPT, Pen, White Board	T1
42	4	Doubt And revision			NIL	CO1, CO3, CO5 & CO7	PPT, Pen, White Board	T1

43	5	**Database Programming**: Introduction,	Introduction	https://drive.google.com/drive/folders/1acFXe12dKEJIvQNBpfEiWPw92tWpp9LN	NIL	CO1, CO5, CO6 & CO7	PPT, Pen, White Board	T1
44	5	Database Programming	. Databases and Python Basic Database Operations and SQL		NIL	CO1, CO5, CO6 & CO7	PPT, Pen, White Board	T1
45	5	Python Database Application Programmer's Interface	. Introduction Python Database Application Programmer's Interface (DB-API) Module Attributes		NIL	CO1, CO5, CO6 & CO7	PPT, Pen, White Board	T1
46	5	Python Database Application Programmer's Interface	. Connection Objects Cursor Objects		NIL	CO1, CO3, CO5 & CO7	PPT, Pen, White Board	T1
45	5	Python Database Application Programmer's Interface	. Databases and Python: Adapters Examples of Using Database Adapters		NIL	CO1, CO3, CO5 & CO7	PPT, Pen, White Board	T1
47	5	(DB-API)	. Implementation of DB-API		NIL	CO1, CO5, CO6 & CO7	PPT, Pen, White Board	T1
48		Object Relational Managers (ORMs) Related Modules	. Python and ORMs SQL Alchemy ORM Example Database-Related Modules		NIL	CO1, CO3, CO5 & CO7	PPT, Pen, White Board	T1
		**Python with IOT (Rasbeerypi)	. Introduction to Rasbeerypi		NIL	CO1, CO5, CO6 & CO7	PPT, Pen, White Board	T1
49		*TIPS FOR UNIVERSITY EXAM PREPARATION						

		-

XII. Mapping Course Outcomes Leading to the Achievement of Program Outcomes and Program Specific Outcomes:

Course Outcomes	Program Outcomes (PO)												Program Specific Outcomes (PSO)		
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
CO1	3	3	3	3	3	-	-	1	2	-	2	1	3	3	2
CO2	3	3	3	3	3	-	-	1	2	-	2	1	3	3	2
CO3	3	3	3	3	3	-	-	1	2	-	2	1	3	3	2
CO4	3	3	3	3	3	-	-	1	2	-	3	1	3	3	2
CO5	3	3	3	3	3	-	-	1	2	-	3	1	3	3	2
AVG	3.0	3.0	3.00	3.0	3.0			1.0	2.0		2.40	1.00	3.00	3.00	2.00

1: Slight (Low) **2: Moderate (Medium)** **3: Substantial (High)** **- : None**

Descriptive Questions

X. QUESTION BANK DESCRIPTIVE QUESTIONS:

Short Answer Questions-

UNIT-I

S.NO	QUESTION	BLOOMSTaxonomy
1.	Explain the difference between compiled and interpreted	L2: UNDERSTAND
2.	What are mutable and immutable types?	L1: REMEMBER
3.	What happens if a semicolon (;) is placed at the end of a Python	L1: REMEMBER
4.	Define dictionary in Python	L1: REMEMBER
5.	Explain the features of tuple data structure	L2: UNDERSTAND

Long Answer Questions-

S.NO	QUESTION	BLOOMS Taxonomy
1.	Explain about the need for learning python programming and	L2: UNDERSTAND
2.	Write in brief about the applications of Python.Give	L2: UNDERSTAND
3.	Explain the following operators in python with appropriate	L2: UNDERSTAND
4.	Explain about methods in Lists of Python with appropriate	L2: UNDERSTAND
5.	Give a comparison between lists, tuples, dictionaries and sets.	L5: EVALUATE

UNIT-2

Short Answer Questions

S.NO	QUESTION	BLOOMS Taxonomy
1.	Define File Objects?	L1: REMEMBER
2.	What is meant Exceptions as Strings?	L1: REMEMBER
3.	Define File Built-in Function [open()]?	L1: REMEMBER
4.	Can a Python function return multiple values? If yes, how it	L2: UNDERSTAND
5.	List out different File Built-in Methods	L2: UNDERSTAND

Long Answer Questions-

S.NO	QUESTION	BLOOMS Taxonomy
1.	What type of parameter passing is used in Python? Justify your	L2: UNDERSTAND
2.	Write a Python program that overloads + operator, to add two	L2: UNDERSTAND
3.	What are the two ways of importing a module? Which one is	L2: UNDERSTAND
4.	Explain in brief about Packages?	L2: UNDERSTAND
5.	Explain how to implement inheritance in Python.	L2: UNDERSTAND

UNIT-3

Short AnswerQuestions-

S.NO	QUESTION	BLOOMSTaxonomy
1.	Describe the terms Threads in python?	L2: UNDERSTAND
2.	Describe Special Symbols and Characters?	L2: UNDERSTAND
3.	Describe Terms Processes in python?	L2: UNDERSTAND
4.	Define Threading Module?	L2: UNDERSTAND
5.	Define Regular Expressions?	L2: UNDERSTAND

Long AnswerQuestions-

S.NO	QUESTION	BLOOMS Taxonomy
1.	Explain the methods that are used to synchronize threads?	L2: UNDERSTAND
2.	What are regular expressions? How to find whether an email	L2: UNDERSTAND
3.	What is multithreading? Discuss about starting a new thread.	L2: UNDERSTAND
4.	Explain in detail about Global Interpreter Lock with example?	L2: UNDERSTAND
5.	Explain in detail about Res and Python Multithreaded	L2: UNDERSTAND

UNIT-4

Short AnswerQuestions-

S.NO	QUESTION	BLOOMS Taxonomy
1.	Describe Building CGI Application.	L2: UNDERSTAND
2.	Define CGI-Helping Servers Process Client Data.	L2: UNDERSTAND
3.	What is tkinter TK ()?	L2: UNDERSTAND
4.	What is the best GUI for Python.	L2: UNDERSTAND
5.	How tkinter applications can be freezed?	L2: UNDERSTAND

Long AnswerQuestions-

S.NO	QUESTION	BLOOMS Taxonomy
1.	Explain about Radio button widget in tkinter. How to create	L2: UNDERSTAND
2.	Write a Python program that creates a GUI with a textbox, Ok	L2: UNDERSTAND
3.	Explain in detail about Web (HTTP) Servers.	L2: UNDERSTAND
4.	Write a program for basic web browser using Tkinter which	L3: APPLY
5.	Explain with an example about Wed Surfing with Python?	L2: UNDERSTAND

UNIT-5

Short AnswerQuestions-

S.NO	QUESTION	BLOOMS Taxonomy
1.	Define usage of following Type Object.	L1: REMEMBER
2.	What is meant by frameworks?	L2: UNDERSTAND
3.	Define Databases and Python Adapters	L1: REMEMBER
4.	What is database schema?	L2: UNDERSTAND
5.	What is the use of cursor.getrowid() method .	L2: UNDERSTAND

Long AnswerQuestions-

S.NO	QUESTION	BLOOMSTaxonomy
1.	Write the syntax to open a database in python?	L2: UNDERSTAND
2.	Write the syntax to execute database queries to perform the	L2: UNDERSTAND
3.	Explain in detail about Object Relational Managers?	L2: UNDERSTAND
4.	Discuss about Python Database Application Programmer's	L2: UNDERSTAND
5.	Explain following connection objects.	L2: UNDERSTAND

Descriptive Questions

1. Define python? List the standard data types of python?
2. Define variable in python and list the rules of python variables?
3. Write a python program to create variables in terms of integer, float and string?
4. Write a python code to demonstrate type conversions using int (), float () and str ()?
5. What are the different types of operators used to evaluate Boolean expression?
6. Write a python program to find biggest of two numbers using conditional if?
7. . Write a python function using with parameter and return type?
8. Define python? List the standard data types of python?
9. Define variable in python and list the rules of python variables?
10. Write a python program to create variables in terms of integer, float and string?
11. Write a python code to demonstrate type conversions using int (), float () and str ()?
12. What are the different types of operators used to evaluate Boolean expression?
13. Write a python program to find biggest of two numbers using conditional if?
14. . Write a python function using with parameter and return type?
15. Write a python program using function to the print the value of x as local and global?
16. Define file and explain the two categories of files?
17. How to import a module from a package show with an example?

18. Define Exception? List any 6 types of exception?
19. How to rename a module in python and write the syntax and program?
20. Define python? List the standard data types of python?
21. Define variable in python and list the rules of python variables?
22. Write a python program to create variables in terms of integer, float and string?
23. Write a python code to demonstrate type conversions using int (), float () and str ()?
24. What are the different types of operators used to evaluate Boolean expression?
25. Write a python program to find biggest of two numbers using conditional if?
26. . Write a python function using with parameter and return type? [5+5+4]
27. Write a python program using function to the print the value of x as local and global?
28. Define file and explain the two categories of files?
29. How to import a module from a package show with an example?
30. Define Exception? List any 6 types of exception?
31. How to rename a module in python and write the syntax and program?

OBJECTIVE QUESTIONS

UNIT 1

1. What Is The Default Return Value For A Function That Does Not Return Any ValueExplicitly?

2. Which Of The Following Items Are Present In The FunctionHeader?
A. function name **B. function name and parameter list**
C. parameter list **D.** return value

3.*What Will Be The Output Of The Following Code Snippet?*

```
a=[1,2,3,4,5,6,7,8,9]

print(a[::2])
```

A. [1,2] **B.** [8,9] **C. [1,3,5,7,9]** **D.** [1,2,3]

4. What Will Be The Output Of The Following CodeSnippet?

```
a=[1,2,3,4,5]
print(a[3:0:-1])
```

A.Syntax error . **[4, 3,2]** **C.**[4,3] **D.** [4, 3, 2,1]

5. *What Will Be The Output Of The Following Code?*

```
class Test:
    definit(self, s): self.s
        = s
     def print(self):
        print(s)
a = Test("Python
```

A. The program gives an error because there is no constructor for classTest.
B. Signature for the print method is incorrect, so an error isthrown.
C. The correct output is.
D. The above code will execute correctly on changing print(s) toprint(self.s).

Q-6What Will Be The Output Of The Following Code?

```
class Test:
    definit(self, s): self.s
       = s
     def print(self):
        print(self.s)
msg = Test()
```

A. The program has an error because class Test does not have aconstructor.
B. The above code produces an error because the definition of print(s) does notinclude
. **C.** It executes successfully but prints nothing.
D. **The program has an error because of the constructor call is made without an argument**.
7 . Wagner–Fischerisa_____algorithm. (**Dynamic programming)**
8. Wagner–Fischer algorithm is usedtofind_____(**Edit distance between two strings)**
9. What is the edit distance between the strings "abcd" and "acbd" when theallowed operations are insertion, deletionand substitution?_____**(2)**

10.What will be theoutput? (2, 4)

```
1. >>>t=(1,2,4,3)
2. >>>t[1:3]
```

UNIT 2

1.To open a file c:\scores.txt for reading, we use

a) infile = open("c:\scores.txt", "r") **b) infile = open("c:\\scores.txt", "r")**
c) infile = open(file = "c:\scores.txt", "r") d) infile = open(file = "c:\\scores.txt", "r")

2.What is theoutput?

```
1. f = None
2.  for i in range (5):
3.     with open("data.txt", "w") as f:
4.         if i> 2:
5.            break
6. print(f.closed)
```

a)True b)False c)None d) Error

3.Can one block of except statements handle multiple exception?
a**) yes, like except TypeError, SyntaxError [,…].**
b) yes, like except [TypeError, SyntaxError].
c) no
d) none of the mentioned

4.Is the following code valid?

```
try:
# Do something
except:
# Do something
finally:
```

```
# Do something
```

a) no, there is no such thing as finally **b) no, finally cannot be used with except**
c) no, finally must come beforeexcept d) yes

5. All modular designs are because of a top-down design process? True or False?
a) True b) False

6. The readlines() method returnsa listof_____ **Answer: Lines**

7.Program code making use of a given module iscalleda_____o the module. **Answer:Client**

8._____is a string literal denoted by triple quotes for providing th specifications of certain programelements. **Answer:Docstring**

10.What will be theoutput?_____**(2, 4)**

```
>>>t=(1,2,4,3)
>>>t[1:3]
```

UNIT 2

1.To open a file c:\scores.txt for reading, we use
a) infile = open("c:\scores.txt", "r") **b) infile = open("c:\\scores.txt", "r")**
c) infile = open(file = "c:\scores.txt", "r") d) infile = open(file = "c:\\scores.txt", "r")

2.What is theoutput?

```
f = None
for i in range (5):
    with open("data.txt", "w") as f:
        if i> 2:
            break
print(f.closed)
```

a)True b)False c)None d) Error

3.Can one block of except statements handle multiple exception?
a**) yes, like except TypeError, SyntaxError [,…].**
b) yes, like except [TypeError, SyntaxError].
c) no
d) none of the mentioned

4.Is the following code valid?

```
try:
# Do something
except:
# Do something
finally:
# Do something
```

a) no, there is no such thing as finally **b) no, finally cannot be used with except**
c) no, finally must come beforeexcept d) yes

5. All modular designs are because of a top-down design process? True or False?
a) True b) False

9. The readlines() method returnsa listof_____ **Answer: Lines**

10. P the module. **Answer:Client**
ogram code making use of a given module iscalleda_____o specifications of certain

11. _____is a string literal denoted by triple quotes for providing the programelements. **Answer:Docstring**

9._____exceptions are raised as a result of an error in opening aparticular file. **Answer:IOError**

10. Methods of a class that provide access to private members of the class are calledas _____ and _____ **Answer:getters/setters**

UNIT III

1. Which module in Python supports regular expressions?
a) re b) regex c) pyregex d) none of the mentioned
2. Which of the following creates a pattern object?
a) re.create(str) b) re.regex(str) **c) re.compile(str)** d) re.assemble(str)
3. What does the function re.match do?
a) **matches a pattern at the start of the string** b) matches a pattern at any position in the string

c) such a function does not exist d) none of the mentioned 4. Which of the following functions clears the regular expression cache?
a) re.sub() b) re.pos() **c) re.purge()** d) re.subn()
5 What is the output of the line of code shown below?
re.split('\W+', 'Hello, hello, hello.')
a) [‘Hello’, ‘hello’, ‘hello.’] b) [‘Hello, ‘hello’, ‘hello’]
c) [‘Hello’, ‘hello’, ‘hello’, ‘.’] **d) [‘Hello’, ‘hello’, ‘hello’, ”]**
6. The character Dot (that is, ‘.’) in the default mode, matches any character other than _____(newline)
7.The expression a{5}willmatch_____characters with the previousregular expression.(exactly 5)
8. _____functions matches a pattern at any position in thestring(re.search)

9. In the functions re.search.start(group) and re.search.end(group), if the argument groupsnot specified, itdefaultsto_____(Zero)
10._____functions does not accept anyargument(re.purge)

UNITIV

1. How do you create awindow??
a) window = newWindow() b) window = Window()
c) window=Frame() **d) window =Tk()**
2. How do you create aframe?
a) frame=newWindow() b) frame = Window()
c) frame=Frame() d) frame = Tk()
3. How do you create an eventloop??
a) window.loop() b) window.main()**c) window.mainloop()** d) window.eventloop()
4. How do you create a canvas under parent frame1 with background color white and foregroung color green?
a) Canvas(frame1, bg = "white", fg ="green")
b) Canvas(frame1, bg = "white", fg = "green", command =processEvent)
c) Canvas(frame1, bg = "white", command =processEvent)
d) Canvas(frame1, fg = "green", command =processEvent)
5. To display an error dialog named "Variable is not assigned",use_____
a) tkinter.messagebox.showinfo("showinfo", "Variable is notassigned")
b) tkinter.messagebox.showwarning("showwarning", "Variable is notassigned")
c) tkinter.messagebox.showerror("showerror", "Variable is notassigned")
d) tkinter.messagebox.askyesno("ashyesno", "Variable is notassigned")

6. grid()method_____
7. w=Canvas(_____) Answer : master,option=value
8. Listbox)_____Answer : offers a list to the user from which the user canaccept any number ofoptions.
9. CGI standsfor_____
10. Module used for GUI and webprogramming_____
1. Which method is used to retrieve the executed database function or stored procedure result in Python

a)**cursor.stored_results()**b)cursor.get_results()c)cursor.fetch_results()

2. Which method of cursor class is used to get the number of rows affected after any of the insert/update/delete database operation executed from Python
a)cursor.rowcount **b)**cursor.getaffectedcount c)cursor.rowscount
3. Which method is used to Commit pending transaction to the database in Python?
a)**connection.commit()**b.cursor.commit()
4.Mandatory arguments required to connect any database from Python a)Username, Password, Hostname, Database Name, Port.
b)Username, Password, Hostname
c) Username, Password, Hostname, Database Name
5.Exception raised when the relational integrity of the database is affected in Python
a)IntegrityFailError **b)IntegrityError** c)IntegrityViolationError
6.ORMsstands_____(Object relation models)
7.DB-API standsfor_____
8. Relational databases are the most widely used type of database, storing informationas tables containing a number ofrows.(TRUE/FALSE)
9._____method of cursor class is used to fetch limited rows from thetable (cursor.fetchmany(SIZE))
10_____method of cursor class is used to get the number of rows affected after anyof the insert/update/delete database operation executed from Python(cursor.rowcount**)**

GATE QUESTIONS
NOT Related

Websites:
https://www.python.org/https://pythonprogramming.net
/
https://www.edureka.co/blog/python-programming-languagehttps://www.programiz.com

XIV. LIST OF TOPICS FOR STUDENTS'SEMINARS

1. PythonBasics
2. Lists, andTuples