PYTHON PROGRAMMING

Subject Code: **CS721PE**

Regulations : R16 - JNTUH

Class: IV Year B.Tech CSE I Semester

Department of Computer Science and Engineering

BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY Ibrahimpatnam - 501 510, Hyderabad

PYTHON PROGRAMMING (CS721PE)
COURSE PLANNER

I.COURSE OVERVIEW:

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.

II.PRE-REQUISITES:

Experience with a high level language (C/C++, Java, MATLAB) is suggested. Prior knowledge of a scripting language (Perl, UNIX/Linux shells) and Object-Oriented concepts is helpful but not mandatory.

III. COURSE OBJECTIVIES:

- To be able to introduce core programming basics and program design with functions using Python programming language.
- To understand a range of Object-Oriented Programming, as well as in-depth data and information processing techniques.
- To understand the high-performance programs designed to strengthen the practical expertise.

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)

S. No.	Course Outcomes (CO)
After completing this course the student must demonstrate the knowledge and ability to:	
CO1	Examine Python syntax and semantics and be fluent in the use of Python flow control and functions.
CO2	Demonstrate proficiency in handling Strings and File Systems.
CO3	Create, run and manipulate Python Programs using core data structures like Lists, Dictionaries and use Regular Expressions.
CO4	Interpret the concepts of Object-Oriented Programming as used in Python.
CO5	Implement exemplary applications related to Network Programming, Web Services and Databases in Python.

V. How Program Outcomes are Assessed:

Program Outcomes (PO)		Level	Proficiency assessed by
PO1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.	2	Assignments, Tutorials, Mock Tests
PO2	**Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	2	Assignments, Tutorials
PO3	**Design/development of solutions**: Design solutions for complex engineering problems and design 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, Tutorials, Mock Tests
PO4	**Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.	3	Assignments, Tutorials, Mock Tests
PO5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.	2	Assignments, Tutorials, Mock Tests
PO6	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.	3	Assignments, Tutorials, Mock Tests
PO7	**Environment and sustainability**: Understand the impact of the professional engineering solutions in 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 of the engineering practice.	-	-
PO9	**Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.	-	-
PO10	**Communication**: Communicate effectively on complex engineering activities with the engineering community 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.	-	-

Program Outcomes (PO)		Level	Proficiency assessed by
PO11	**Project management and finance**: Demonstrate knowledge and understanding of the engineering 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	Assignments, Tutorials, Mock Tests
PO12	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.	-	-

VI.HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED:

Program Specific Outcomes (PSO)		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, Tutorials, Mock Tests
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	Assignments, Tutorials
PSO3	**Foundations of Software development:** The ability to grasp the software development lifecycle and methodologies of software 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.	3	Assignments, Tutorials, Mock Tests

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

VII. 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: 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 App lication Programmer's Interface (DB-API), Object Relational Managers (ORMs), Related Modules

Textbook

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

VIII. LESSON PLAN:

SNO	Week No	Topic to be covered	Learning Objectives	Teachingmethodology	References
		UNIT-I			
1	**1**	Introduction to python programming	Explain python programming introduction	Chalk and Talk	T1
2		Python Basics	Define basics in python	Chalk and Talk & PPT	
3		Objects- Python Objects	Understand objects in python	Chalk and Talk	
4		Standard Types	Illustrate standard data types	Chalk and Talk	
5		Other Built-in Types, Internal Types	Define internal types	Chalk and Talk	
6	**2**	Standard Type Operators, Standard Type Built-in Functions	Explain Standard Type Operators, Standard Type Built-in Functions	Chalk and Talk	

7		Categorizing the Standard Types,Unsupported Types	Categorizing the Standard Types,Unsupported Types	Chalk and Talk	
8		Numbers - Introduction to Numbers, Integers	Explain Numbers , Integers	Chalk and Talk	
9		Floating Point Real Numbers, Complex Numbers	Discuss Floating Point Real Numbers, Complex Numbers	Chalk and Talk	
10		Operators, Built-in Functions, Related Modules	Discuss Operators, Built-in Functions, Related Modules	Chalk and Talk	
11		Sequences - Strings	Explain Sequences - Strings	Chalk and Talk& PPT	
12		Lists, and Tuples	Describe Lists, and Tuples	Chalk and Talk& PPT	
13	**3**	Mapping and Set Types	Understand Mapping and Set Types	Chalk and Talk& PPT	
		UNIT-II			
14		FILES: File Objects	Understand FILES: File Objects	Chalk and Talk& PPT	
15		File Built-in Function [open()]	Describe File Built-in Function [open()]	Chalk and Talk& PPT	
16		File Built-in Methods	Define File Built-in Methods	Chalk and Talk& PPT	
17		File Built-in Attributes, Standard Files	Explain File Built-in Attributes, Standard Files	Chalk and Talk& PPT	
18		Command-line Arguments	Analyze Command-line Arguments	Chalk and Talk& PPT	T1
19	**4**	File System, File Execution	Describe File System, File Execution	Chalk and Talk& PPT	
20		Persistent Storage Modules	Distinguish Persistent Storage Modules	Chalk and Talk& PPT	
21		**MOCK TEST I**		Chalk and Talk& PPT	
22		Related Modules Exceptions: Exceptions in Python, Detecting and Handling Exceptions	Related Modules Exceptions: Exceptions in Python, Detecting and Handling Exceptions	Chalk and Talk	

23		Context Management,	Context Management,	Chalk and Talk	
24		*Exceptions as Strings, Raising Exceptions	*Exceptions as Strings, Raising Exceptions	Chalk and Talk	
25		Assertions, Standard Exceptions	Assertions, Standard Exceptions	Chalk and Talk	
26	**5**	*Creating Exceptions, Why Exceptions? Why Exceptions at All?	*Creating Exceptions, Why Exceptions? Why Exceptions at All?	Chalk and Talk	
27		**BRIDGE CLASS 1**		Chalk and Talk	
28		Exceptions and the sys Module	Define Exceptions and the sys Module	Chalk and Talk	
29		Related Modules Modules: Modules and Files	Understand Related Modules Modules: Modules and Files	Chalk and Talk	
30		Namespaces, Importing Modules	Exaplin Namespaces, Importing Modules	Chalk and Talk	
31	**6**	Importing Module Attributes	Understand Importing Module Attributes	Chalk and Talk	
32		**BRIDGE CLASS 2**		Chalk and Talk	
33		Module Built-in Functions, Packages, Other Features of Modules	Apply Module Built-in Functions, Packages, Other Features of Modules	Chalk and Talk	
		UNIT III			
34		Regular Expressions: Introduction	Explain Regular Expressions: Introduction	Chalk and Talk	
35		Special Symbols and Characters	Understand Special Symbols and Characters	Chalk and Talk	
36	**7**	Res and Python Multithreaded Programming: Introduction	Exaplain Res and Python Multithreaded Programming: Introduction	Chalk and Talk	
37		Threads and Processes	Distinguish between Threads and Processes	Chalk and Talk	
38		**BRIDGE CLASS 3**		Chalk and Talk	T1
	8	**MID 1 EXAMS**			
39		Python	Exaplin Python	Chalk and Talk	
40	**9**	Threads and the Global Interpreter Lock	Understand Threads and the Global Interpreter Lock	Chalk and Talk	
41		Thread Module	Apply Thread Module	Chalk and Talk	
42		Threading Module, Related Modules	Discuss Threading Module, Related Modules	Chalk and Talk	

43		**BRIDGE CLASS 4**		Chalk and Talk	
		UNIT IV			
44	**10**	GUI Programming: Introduction	Explain GUI Programming: Introduction	Chalk and Talk,PPT	
45		GUI Programming: Introduction	Explain GUI Programming: Introduction	Chalk and Talk,PPT	
46		Tkinter and Python Programming	Understand Tkinter and Python Programming	Chalk and Talk,PPT	
47		Tkinter and Python Programming	Understand Tkinter and Python Programming	Chalk and Talk,PPT	
48		Brief Tour of Other GUIs	Define Brief Tour of Other GUIs	Chalk and Talk,PPT	
49	**11**	**BRIDGE CLASS 5**		Chalk and Talk,PPT	
50		Brief Tour of Other GUIs	Define Brief Tour of Other GUIs	Chalk and Talk,PPT	
51		Related Modules and Other GUIs WEB Programming: Introduction	Understand Related Modules and Other GUIs WEB Programming: Introduction	Chalk and Talk,PPT	
52		WEB Programming: Introduction	Describe WEB Programming: Introduction	Chalk and Talk,PPT	
53		WEB Programming:	Describe WEB Programming:	Chalk and Talk,PPT	T1
54	**12**	Wed Surfing with Python	Discuss Wed Surfing with Python	Chalk and Talk,PPT	
55		**BRIDGE CLASS 6**		Chalk and Talk,PPT	
56		Creating Simple Web Clients	Apply Creating Simple Web Clients	Chalk and Talk,PPT	
57		Creating Simple Web Clients	Apply Creating Simple Web Clients	Chalk and Talk,PPT	
58		Advanced Web Clients	Elaborate Advanced Web Clients	Chalk and Talk,PPT	
59	**13**	CGI-Helping Servers Process Client Data	Explain CGI-Helping Servers Process Client Data	Chalk and Talk,PPT	
60		**MOCK TEST II**		Chalk and Talk,PPT	
61		Building CGI Application Advanced CGI	Building CGI Application Advanced CGI	Chalk and Talk,PPT	
62		Web (HTTP) Servers	Web (HTTP) Servers	Chalk and Talk,PPT	
63		Revision	Revision	Chalk and Talk,PPT	
	UNIT V				T1

64		**BRIDGE CLASS 7**		Chalk and Talk,PPT	
65		**DatabaseProgramming**: Introduction,	**Explain Database Programming**: Introduction,	Chalk and Talk,PPT	
66	**14**	Database Programming	Exaplin Database Programming	Chalk and Talk,PPT	
67		Python Database Application Programmer's Interface	Discuss Python Database Application Programmer's Interface	Chalk and Talk,PPT	
68		**BRIDGE CLASS 8**		Chalk and Talk,PPT	
69		Python Database Application Programmer's Interface	Define Python Database Application Programmer's Interface	Chalk and Talk,PPT	
70		Python Database Application Programmer's Interface	define Python Database Application Programmer's Interface	Chalk and Talk,PPT	
71	**15**	(DB-API)	Understand (DB-API)	Chalk and Talk,PPT	
72		(DB-API)	Undersatnd (DB-API)	Chalk and Talk,PPT	
73		Object Relational Managers (ORMs)	Exaplain Object Relational Managers (ORMs)	Chalk and Talk,PPT	
74		Object Relational Managers (ORMs)	Explain Object Relational Managers (ORMs)	Chalk and Talk,PPT	
75		Related Modules	Understand Related Modules	Chalk and Talk,PPT	
76	**16**	Related Modules	Understand Related Modules	Chalk and Talk,PPT	
77		**BRIDGE CLASS 9**		Chalk and Talk,PPT	
78		**BRIDGE CLASS 10**		Chalk and Talk,PPT	
	17	**MID II**			

Textbook

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

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

Course	Program Outcomes (PO)												Program Specific Outcomes (PSO)		
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**

CO1	3	2	2	3	2	2	-	-	-	-	-	2	2	2	1
CO2	3	2	2	2	2	2	-	-	-	-	-	1	1	2	1
CO3	3	3	3	3	2	2	-	-	-	-	-	1	1	2	1
CO4	2	2	3	2	3	3	-	-	-	-	-	2	1	2	2
CO5	1	2	2	3	2	2	-	-	-	-	-	1	2	2	2
AVG	2	2	2	3	2	2	-	-	-	-	-	1	1	2	1

X. QUESTION BANK

DESCRIPTIVE QUESTIONS:

UNIT-I

Short Answer Questions-

S.NO	QUESTION	BLOOMS Taxonomy
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 Answer Questions-

S.NO	QUESTION	BLOOMS Taxonomy
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 Answer Questions-

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 Answer Questions-

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 Answer Questions-

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 Answer Questions-

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 Answer Questions-

S.NO	QUESTION	BLOOMS Taxonomy
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

OBJECTIVE QUESTIONS

UNIT 1

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

2. Which Of The Following Items Are Present In The Function Header?

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 Code Snippet?

```
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:
    def __init__(self, s):
        self.s = s
    def print(self):
        print(s)
a = Test("Python
Class") a.print()
```

A. The program gives an error because there is no constructor for class Test.

B. Signature for the print method is incorrect, so an error is thrown.

C. The correct output is .

D. The above code will execute correctly on changing print(s) to print(self.s).

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

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

A. The program has an error because class Test does not have a constructor.

B. The above code produces an error because the definition of print(s) does not include

. **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–Fischer is a _____ algorithm. (**Dynamic programming)**

8. Wagner–Fischer algorithm is used to find _____ (**Edit distance between two strings)**

9. What is the edit distance between the strings “abcd” and “acbd” when the allowed operations are insertion, deletion and substitution?_____**(2)**

10.What will be the output?_____ **(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 the output?

```
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 before except 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 list of_____ **Answer: Lines**

7.Program code making use of a given module is called a _____ of the module. **Answer:Client**

8._____ is a string literal denoted by triple quotes for providing the specifications of certain program elements. **Answer:Docstring**

9._____ exceptions are raised as a result of an error in opening a particular file. **Answer: IOError**

10.Methods of a class that provide access to private members of the class are called as _____ 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} will match _____ characters with the previous regular expression.(exactly 5)
8. _____ functions matches a pattern at any position in the string(re.search)

9. In the functions re.search.start(group) and re.search.end(group), if the argument groups not specified, it defaults to _____(Zero)
10._____ functions does not accept any argument(re.purge)

UNIT IV

1. How do you create a window??
a) window = newWindow() b) window = Window()
c) window = Frame() **d) window = Tk()**
2. How do you create a frame?
a) frame = newWindow() b) frame = Window()
c) frame = Frame() d) frame = Tk()
3. How do you create an event loop??
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 not assigned")
b) tkinter.messagebox.showwarning("showwarning", "Variable is not assigned")
c) tkinter.messagebox.showerror("showerror", "Variable is not assigned")
d) tkinter.messagebox.askyesno("ashyesno", "Variable is not assigned")
6. grid() method_____
7. w = Canvas(_____) Answer : master, option=value
8. Listbox)_____Answer : offers a list to the user from which the user can accept any number of options.
9. CGI stands for_____
10. Module used for GUI and web programming_____

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.ORMs stands _____(Object relation models)

7.DB-API stands for_____

8.Relational databases are the most widely used type of database, storing information as tables containing a number of rows.(TRUE/FALSE)

9._____method of cursor class is used to fetch limited rows from the table (cursor.fetchmany(SIZE))

10_____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 (cursor.rowcount**)**

GATE QUESTIONS

Not Related

XI.WEBSITES:

https://www.python.org/

https://pythonprogramming.net/

https://www.edureka.co/blog/python-programming-language

https://www.programiz.com

XII.EXPERT DETAILS

1. **Wesley J. Chun**
2. https://www.innoappstech.com/?utm_medium=nancy&utm_source=top+python+programmers++/+quora
3. https://www.valuecoders.com/?utm_medium=nancy&utm_source=top+python+programmers++/+quora
4. https://www.pixelcrayons.com/?utm_medium=nancy&utm_source=top+python+programmers++/+quora
5. **Guido van Rossum**

XIII.JOURNALS

1. **Programming with Python DOI:** 10.1109/MITP.2005.120 **Publisher:** IEEE
2. Python Power **DOI:** 10.1109/MCSE.2014.26 **Publisher:** IEEE
3. **Exploration of teaching method of Python Programming based on the case of technical problem** DOI: **10.1109/ICCSE.2017.8085563** Publisher: **IEEE**

XIV.LIST OF TOPICS FOR STUDENTS' SEMINARS

1. Python Basics
2. Lists, and Tuples

FILES:

Exceptions in Python, Detecting and Handling Exceptions

GUI Programming

WEB Programming:

3. Creating Simple Web Clients
4. Python Database Application Programmer's Interface

XV.CASE STUDIES / PROJECTS

Dice Rolling Simulator
Guess the Number
TextBased Adventure Game
Mad Libs generator
Hangman