A

COURSE FILE

ON

“PYTHON PROGRAMMING”

Submitted by

U.Mahender
Assistant Professor

In the Department of
Computer Science and Engineering

CMR ENGINEERING COLLEGE

KANDLAKOYA (V), MEDCHAL (M), R.R.DIST.

A.Y-2023-24

CONTENTS OF COURSE FILE:

1. Department vision & mission

2. List of PEOs and POs PSO’s

3. List of Cos (Action verbs as per Bloom’s Taxonomy)

4. Syllabus copy and Suggested/Reference Books

5. Individual Time table

6. Session plan/Lesson Plan

7. Session execution log

8. Lecture Notes (Handwritten)

9. Assignment Questions (original or xerox of mid-1 and mid-2) with Assignments sample.

10. Mid-exam question papers with (xerox of mid-1 and mid-2) Script sample.

11. Scheme of evaluation

12. Mapping of COs with POs and PSOs

13. COs, POs, PSOs Justification

14. Attainment of COs, POs and PSOs (Excel sheet)

15. Previous question papers

16. PowerPoint Presentation (PPTs)

17. Innovative teaching methods

18. References (Websites/URLs/e- Resources)

Submitted By

U.Mahender

Assistant. Prof (CSE)

1. DEPARTMENT VISION & MISSION

Vision:

To produce globally competent and industry-ready graduates in Computer Science & Engineering by imparting quality education with the know-how of cutting-edge technology and holistic personality.

Mission:

1. To offer high-quality education in Computer Science & Engineering in order to build core competence for the graduates by laying a solid foundation in Applied Mathematics and program framework with a focus on concept building.

2. The department promotes excellence in teaching, research, and collaborative activities to prepare graduates for a professional career or higher studies.

3. Creating an intellectual environment for developing logical skills and problem-solving strategies, thus developing, an able and proficient computer engineer to compete in the current global scenario.

2. LIST OF PEOs, POs AND PSOs

2.1 Program Educational Objectives (PEO):

PEO 1: Excel in professional career and higher education by acquiring knowledge of mathematical computing and engineering principles.

PEO 2: To provide an intellectual environment for analyzing and designing computing systems for technical needs.

PEO 3: Exhibit professionalism to adapt current trends using lifelong learning with legal and ethical responsibilities.

PEO 4: To produce responsible graduates with effective communication skills and multidisciplinary practices to serve society and preserve the environment.

2.2. Program Outcomes (POs):

Engineering Graduates will be able to satisfy these NBA graduate attributes:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems
2. **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
3. **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
4. **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
5. **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
6. **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
7. **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
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings
10. **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
11. **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
12. **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

2.3 Program Specific Outcomes (PSOs):

PSO1: Professional Skills and Foundations of Software development: Ability to analyze, design and develop applications by adopting the dynamic nature of Software developments.

PSO2: Applications of Computing and Research Ability: Ability to use knowledge in cutting edge technologies in identifying research gaps and to render solutions with innovative ideas.

3. List of CO's(Action verbs as per Bloom's Taxonomy)

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.

CO6. **Develop** the skill of designing Graphical user Interfaces in Python.

4. 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,*ExceptionsasStrings,RaisingExceptions,Assertions,StandardExceptions, *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 Application Programmer’s Interface (DB-API), Object Relational Managers (ORMs), Related Modules

TEXT BOOK:

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

5. INDIVIDUAL TIME TABLE (U.Mahender)

	I	II	III	IV		V	VI	VII
MON		PP-lab(II-A)						
TUE								
WED		PP LAB(II-A)						
THU			PP(II-A)					
FRI			PP(II-A)					
SAT						PP(II-A)		

6. SESSION PLAN/LESSON PLAN

S.NO	Topic (JNTU syllabus)	Sub-Topic	NO. OF LECTURES REQUIRED	Suggested Books	Teaching Methods
UNIT – I					
1	**Python Objects, Numbers & Sequences**	Python basics	**L1**	**T1**	**M1**
2		Python Objects, Standard Types	**L2-L3**	**T1**	**M1**
3		Other Built-in Types, Internal Types	**L4**	**T1**	**M2(PPT)**
4		Standard Type Operators, Standard Type Built-in Functions	**L5-L6**	**T1**	**M2(PPT)**
5		Categorizing the Standard Types, UnsupportedTypes	**L7**	**T1**	**M2**
6		Introduction to Numbers, Integers, Floating Point Real Numbers	**L8**	**T1**	**M2(PPT)**

7		Complex Numbers, Operators, Built-in Functions, Related Modules	**L9**	**T1**	**M1**
8		Sequences - Strings, Lists, and Tuples	**L10-L11**	**T1**	**M1**
9		Mapping and Set Types	**L12**	**T1**	**M1**
UNIT – II					
10	**Files, Exceptions & Modules**	File Objects, File Built-in Function [open()]	**L13**	**T1**	**M2**
11		File Built-in Methods, File Built-in Attributes, Standard Files	**L14**	**T1**	**M2(PPT)**
12		Command-line Arguments, File System, File Execution	**L15-L16**	**T1**	**M1**
13		Persistent Storage Modules, Related Modules	**L17**	**T1**	**M2(NPTEL)**
14		Exceptions in Python, Detecting and Handling Exceptions	**L18**	**T1**	**M2(PPT)**
15		Context Management,*ExceptionsasStrings	**L19**	**T1**	**M1**
16		RaisingExceptions,Assertions	**L20**		

				T1	**M1**
17		Standard Exceptions, Creating Exceptions	**L21**	**T1**	**M2(PPT)**
18		Why Exceptions (Now)?, Why Exceptions at All?, Exceptions and the sys Module, Related Modules	**L22**	**T1**	**M2(PPT)**
19		Modules: Modules and Files, Namespaces	**L23**	**T1**	**M1**
20		Importing Modules, Importing Module Attributes	**L24**	**T1**	**M1**
21		Module Built-in Functions, Packages, Other Features of Modules	**L25**	**T1**	**M1**
UNIT-III					
22	**Regular Expressions & Multithreaded Programming**	Introduction to Regular Expressions, Special Symbols and Characters	**L26**	**T1**	**M1**
23		Res and Python	**L27**	**T1**	**M1**
24		Introduction to Multithreaded programming , Threads and Processes	**L28**	**T1**	**M1**
25		Threads, and the Global Interpreter Lock	**L29**	**T1**	**M2(E-resources)**
26		Thread Module, Threading Module, Related Modules	**L30**	**T1**	**M2(NPTEL)**
UNIT-IV					

27	GUI Programming & Web Programming	Introduction GUI, Tkinter and Python Programming	**L31**	**T1**	**M1**
28		Brief Tour of Other GUIs, Related Modules and Other GUIs	**L32**	**T1**	**M1**
29		Introduction to Web Programming, Wed Surfing with Python	**L33**	**T1**	**M2(PPT)**
30		Creating Simple Web Clients, Advanced Web Clients	**L34**	**T1**	**M2(NPTEL)**
31		CGI-Helping Servers Process Client Data	**L35**	**T1**	**M2(NPTEL)**
32		Building CGI Application Advanced CGI	**L36**	**T1**	**M1**
33		Web (HTTP) Servers	**L37**	**T1**	**M1**
UNIT –V					
34	**DATABASE PROGRAMMING**	Introduction	**L38**	**T1**	**M1**
35		Python Database Application Programmer’s Interface (DB-API)	**L39**	**T1**	**M2(PPT)**
36		Object Relational Managers (ORMs)	**L40-L41**	**T1**	**M2(NPTEL)**
37		Related Modules	**L42**	**T1**	**M1**

METHODS OF TEACHING:

M1: Lecture Method	**M4: Presentation /PPT**	**M7: Assignment**
M2: Demo Method	**M5: Lab/Practical**	**M8: Industry Visit**
M3: Guest Lecture	**M6: Tutorial**	**M9: Project Based**

NOTE:

1. Any Subject in a Semester is supposed to be completed in 55 to 65 periods.
2. Each Period is of 50minutes.
3. Each unit duration & completion should be mentioned in the Remarks column.
4. List of Suggested books can be marked with Codes like T1, T2, R1, R2etc.

7. Session Execution Log:

S no	Units	Scheduled started date	Completed date	Remarks
1	**I**	**19/2/2024**	**11/3/2024**	**Completed**
2	**II**	**13/3/2024**	**12/4/2024**	**Completed**
3	**III**	**13/4/2024**	**1/5/2024**	**Completed**
4	**IV**	**3/5/2024**	**8/6/2024**	**Completed**
5	**V**	**10/6/2024**	**24/6/2024**	**Completed**

8. Lecture Notes (Handwritten)

9. Assignment Questions Along with Sample Assignment Scripts

CMR ENGINEERING COLLEGE
EXPLORE TO INVENT

CMR ENGINEERING COLLEGE
UGC AUTONOMOUS
(Approved by AICTE - New Delhi. Affiliated to JNTUH and Accredited by NAAC & NBA)
Kandlakoya (V), Medchal (M), Medchal - Malkajgiri (D)-501401

NBA NATIONAL BOARD OF ACCREDITATION
NAAC

ASSIGNMENT -I
PYTHON PROGRAMMING
II B.TECH II SEM
Academic year 2023-24

1. a) what is operator? Explain various types of operators in detail with appropriate example. **(CO1)**

b) State and explain the features of Python in details. **(CO1)**

2. a) What is Tuple and Set types in Python statements explain all methods in detail. **(CO1)**

b) Explain String types built-in function in Python **(CO1)**

3. a) What is exception? Explain different types of exceptions**. (CO1)**

b) How does the try-except statement work? Demonstrate with an example python code. **(CO1)**

4. a) Explain the following file built-in functions and methods with clear syntax, description illustration

i. open()
ii. file()
iii. seek()
iv. tell
v. read() **(CO2)**

b) write about importing modules and module attributes with examples. **(CO2)**

5. a) Write a Python program to read a text file in Python and print no of line
and no of unique words. **(CO2)**

b) Write a Python program to append the content of the first file to the second file read the file names as command line arguments **(CO2)**

c) What is persistence storage? explain the following modules with examples**. (CO2)**

a. Marshal b) Shelve c) dbm d) pickle

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Approved by AICTE - New Delhi. Affiliated to JNTUH and Accredited by NAAC & NBA)

Kandlakoya (V), Medchal (M), Medchal - Malkajgiri (D)-501401

ASSIGNMENT -II
PYTHON PROGRAMMING
II B.TECH II SEM
Academic year 2023-24

1. Define multithreading. Explain about threading modules in detail. **(CO 4)**

2. Explain Global Interpreter Lock with an example. **(CO 4)**

3. Explain about GUI in detail with an example. **(CO 4)**

4. Explain CGI with an example**. (CO 4)**

5. Describe in detail Python SQLAlchemy ORM with the help of the Employee role database. **(CO 5)**

10. MID-EXAM QUESTION PAPER ALONG SAMPLE ANSWER SCRIPTS

CMR ENGINEERING COLLEGE
UGC AUTONOMOUS
(Approved by AICTE - New Delhi. Affiliated to JNTUH and Accredited by NAAC & NBA)
Kandlakoya (V), Medchal (M), Medchal - Malkajgiri (D)-501401

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

B. TECH-II -SEM-II MID EXAMINATIONS-I

Subject: Python Programming

Time: 10:00 AM to11:30 AM **Date: 11-5-2023**
Branch: CSE **Marks: 25M**

- -

Note: Question paper contains two parts, Part-A and Part- B.

Part-A is compulsory which carries 10 marks.

Answer all questions in part-A.

Answer anyone full question from each unit.

Each question carries5 marks.

PART-A 5x 2M =10 M

		BTL	CO
1.	Explain built in function in python?	1	1
2.	List out the main differences between list and tuples?	1	3
3.	What is pickling ?	2	2
4.	What is an Exception?	1	4
5.	What is Regular Expression?	1	3

PART-B 4 x 5 M = 20 M

		BTL	CO
6.	a) What is operator? explain various types of operators in details with appropriate example? with program	2	1
7.	a) Write a python program to find given integer is even or odd	2	1
	b) How does try-except statement work? Demonstrate with an example python code? with program example	1	1
8.	What is Tuple and Set types in python statements explain all methods in detail with program?	6	3
9.	Define Module ? Write about importing modules and module attributes with example program?	1	4
10.	Illustrate common regular expression symbols and special characters used in python with program.	1	3
11.	Explain String types and built in function in python with program		

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Approved by AICTE - New Delhi. Affiliated to JNTUH and Accredited by NAAC & NBA)

Kandlakoya (V), Medchal (M), Medchal - Malkajgiri (D)-501401

B. TECH-II(R-22)-SEM-II MID-II EXAMINATIONS

Subject: Python Programming

Time: 10:00 AM to11:30 AM **Date: 04-7-2024**

Branch: CSE,IT **Marks: 30 M**

Note: Question paper contains two parts, Part-A and Part- B.

Part-A is compulsory which carries 10 marks.

Answer all questions in part-A.

Answer any four questions in Part-B

Each question carries5 marks.

PART-A 5x2 M=10 M

1. Describe about the Global Interpreter Lock in python? (CO3)

2.What is thread? What are the methods of Thread class in python? (CO3)

3.What is the need of Tkinter module in python? (CO4)

4. Summarize the purpose of pipe and dot symbols used for pattern matching? (CO3)

5. Write short notes on object relational managers (ORMs). (CO5)

PART-B 4 x 5 M = 20 M

6. What are the differences between processes and threads? Explain with example. (CO3)

7. Write a multithreading program to create two threads, one thread displays even numbers and other thread displays odd numbers from 1 to 20? (CO3)

8. Write a Python GUI program to add buttons in your application using tkinter module. (CO4)

9. What is CGI? Explain the overview of CGI in python. (CO4)

10.What is a Cursor object? Explain various methods and attributes of cursor object? (CO5)

11.Write a python script to perform the following: (CO5)

i) To connect to the MySQL database.

ii) To create the table and insert the data into it.

11. SCHEME OF EVALUATION

MID-I SCHEME OF EVALUATION

PART-A

S.NO	THEORY	MARKS	TOTAL
1	Explain built in function in python?	2	2
2	List out the main differences between list and tuples?	2	2
3	What is pickling ?	2	2
4	What is an Exception?	2	2
5	What is Regular Expression?	2	2

PART-B

S.NO	THEORY	MARKS	TOTAL
6	What is operator? explain various types of operators in details with appropriate example? with program	5	5
7	a) Write a python program to find given integer is even or odd	2.5	5
	b) How does try-except statement work? Demonstrate with an example python code? with program example	2.5	
8	What is Tuple and Set types in python statements explain all methods in detail with program?	5	5
9	Define Module ? Write about importing modules and module attributes with example program?	5	5
10	Illustrate common regular expression symbols and special characters used in python with program.	5	5
11	Explain String types and built in function in python with program	5	5

MID-II SCHEME OF EVALUATION

PART-A

S.NO	THEORY	MARKS	TOTAL
1	Describe about the Global Interpreter Lock in python?	2	2
2	What is thread? What are the methods of Thread class in python?	2	2
3	What is the need of Tkinter module in python?	2	2
4	Summarize the purpose of pipe and dot symbols used for pattern matching?	2	2
5	Write short notes on object relational managers (ORMs).	2	2

PART-B

S.NO	THEORY	MARKS	TOTAL
6	Processes versus Threads. What are the differences between processes and threads?	5	5
7	Write a multithreading program to create two threads, or thread displays even numbers and other thread display odd numbers from 1 to 20?	5	5
8	Write a Python GUI program to add a button in your application using tkinter module	5	5
9	What is CGI?Explain the overview of CGI in python.	5	5
10	What is a cursor object?Explain various methods and attributes of cursor object?	5	5
11	i) To connect to the MySQL database. ii) To create the table and insert the data into it.	5	5

12. Mapping of COs with POs and PSOs

Course Outcomes	**Relationship of Course Outcomes (CO) to Program Outcomes (PO)**													
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3	3	-	-	2	-	-	-	-	-	-	-	2	-
CO2	-	-	2	-	2	-	2	-	-	-	-	-	2	-
CO3	2	2	3	-	3	-	-	-	-	-	-	-	-	1
CO4	3	3	2	-	-	-	-	-	-	-	-	2	-	1
CO5	-	-	2	-	3	-	2	-	-	-	-	-	-	-
CO6	1	1	-	-	-	-	2	-	-	-	-	-	2	-

13. COs, POs, PSOs Justification

CO 1: Examine Python syntax and semantics and be fluent in the use of Python flow control and functions. [remembering]

	Justification
PO1	**Correlated with PO1 strongly** because the students able to know the basic syntax and semantics of python programming language. So, overall the correlation of CO1 to PO1 is good.
PO2	**Correlated with PO2 strongly** because Students will able to analyze the complexity of the problem statements with the help of flow control. So, overall the correlation of CO2 to PO2 is good.
PO5	**Correlated with PO5 moderately** because Students will be able to learn the python software which is used to execute the programs. So, overall the correlation of CO3 to PO5 is moderate.
PSO1	**Correlated with PSO1 is moderately** because Students are able to analyze, design and develop the applications by adopting the various python platform execution environments. So, overall the correlation of CO5 to PSO1 is moderate.

CO 2: Demonstrate proficiency in handling Strings and File Systems. [understanding]

	Justification
PO3	**Correlated with PO3 moderately** because Students will be able to handle with the strings. So, overall the correlation of CO2 to PO3 is moderate.
PO5	**Correlated with PO5 moderately** because Students will be able to learn the python software which is used to execute the programs related to strings and files. So, overall the correlation of CO2 to PO5 is moderate.
PO7	**Correlated with PO7 moderately** because Students will be able to do any work effectively in any environment. So, overall the correlation of CO2 to PO7 is moderate.
PSO1	**Correlated with PSO1 is moderately** because Students are able to analyze, design and develop the applications by adopting the various python platform execution environments. So, overall the correlation of CO5 to PSO1 is moderate.

CO 3: Create, run and manipulate Python Programs using core data structures like Lists, Dictionaries and use Regular Expressions. [Apply].

	Justification
PO1	**Correlated with PO1 moderately** because Students will be able to apply the knowledge of data structures in implementing Lists etc., . So, overall the correlation of CO3 to PO1 is moderate.
PO2	**Correlated with PO2 moderately** because Students will be able to design the solutions for different data structures like Lists, Dictionaries etc., So, overall the correlation of CO3 to PO2 is moderate.

PO3	**Correlated with PO3 strongly** because Students will be able to design the solutions to solve the problems related to. So, overall the correlation of CO3 to PO3 is good.
PO5	**Correlated with PO5 strongly** because Students will be able to learn the creation, run and manipulate the Python Programs. So, overall the correlation of CO3 to PO5 is good.
PSO2	**Correlated with PSO2 is Low** Students will be able to use the regular expression. So, overall the correlation of CO3 to PSO2 is low.

CO 4: Interpret the concepts of Object-Oriented Programming as used in Python. [Analyzing]

	Justification
PO1	**Correlated with PO1 strongly** because Students will be able to apply OOPS concept in the Python. So, overall the correlation of CO4 to PO1 is good.
PO2	**Correlated with PO2 strongly** because Students will be able to design the solutions for different problems related to OOPS Concept. So, overall the correlation of CO4 to PO2 is good.
PO3	**Correlated with PO3 moderately** because Students will be able to design the solutions to various problems. So, overall the correlation of CO3 to PO3 is moderate.
PO12	**Correlated with PO12 moderately** because Students will be able to recognize the need for understanding the various features of OOPS. So, overall the correlation of CO4 to PO12 is moderate.
PSO2	**Correlated with PSO2 is Low** because Students will be able to conduct interpretation of data and provide proper conclusions. So, overall the correlation of CO4 to PSO2 is low.

CO 5: Implement exemplary applications related to Network Programming, Web Services and Databases in Python. (Understanding)

	Justification
PO3	**Correlated with PO3 is Moderately** because Students will be able to generate passwords by applying the knowledge of Network Programming. So, overall the correlation of CO5 to PO3 is moderate.
PO5	**Correlated with PO5 is Strongly** because students will be able to apply the concept of Web services. So, overall the correlation of CO5 to PO8 is low.
PO7	**Correlated with PO7 is Moderately** because Students will be able to do any work effectively in any environment. So, overall the correlation of CO5 to PO9 is moderate.

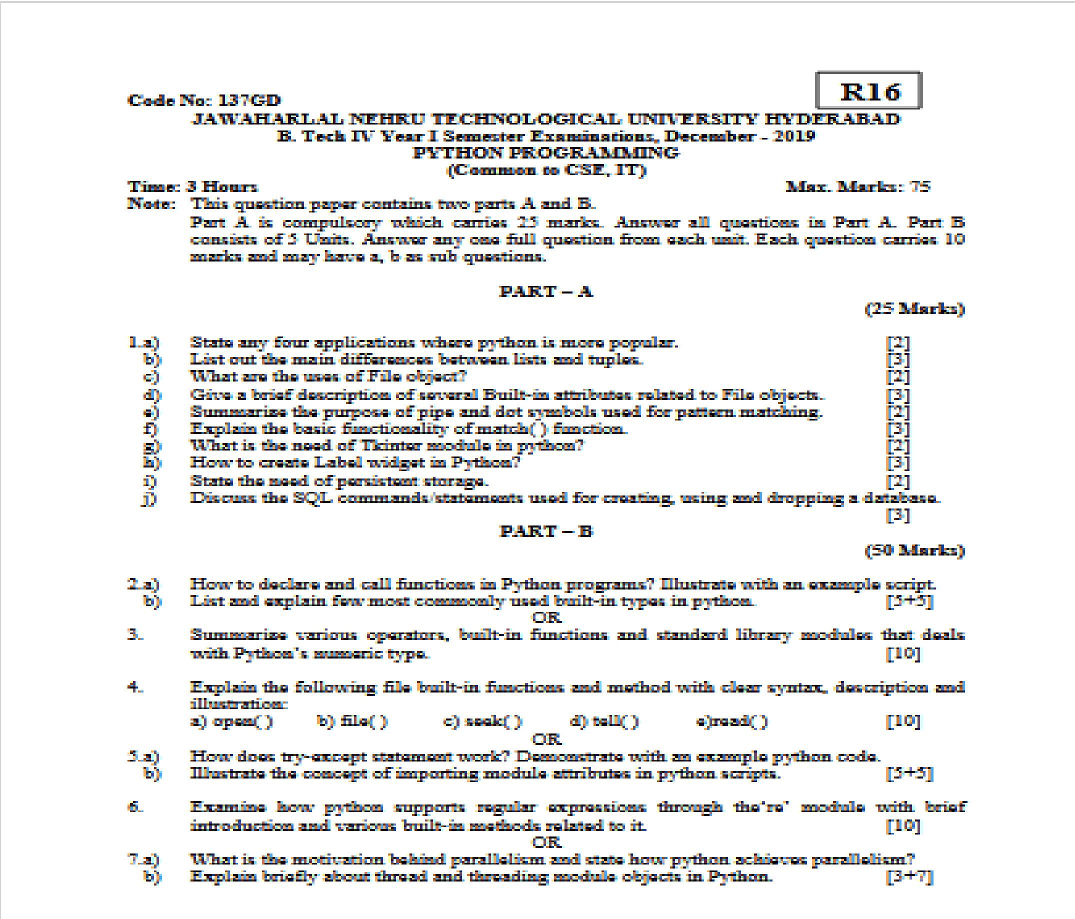
CO 6: Develop the skill of designing Graphical user Interfaces in Python.

	Justification
PO1	**Correlated with PO1 is Low** because Students will be able to understand the concept of Graphical User Interface in Python. So, overall the correlation of CO6 to PO1 is low.
PO2	**Correlated with PO2 is Low** because Students will be able to design the solutions for

	different problems related to GUI. So, overall the correlation of CO6 to PO2 is low.
PO7	**Correlated with PO9 is Moderate** because Students will be able to do any work effectively in GUI environment. So, overall the correlation of CO5 to PO9 is moderate.
PSO1	**Correlated with PSO1 Moderate** because Students will be able to conduct interpretation of data and provide proper conclusions using GUI concept. So, overall the correlation of CO6 to PSO1 is moderate.

14. Attainment of COs, POs and PSOs (Excel Sheet)

15. Previous Year Question Paper



Code No: 137GD **R16**

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
B. Tech IV Year I Semester Examinations, December - 2019
PYTHON PROGRAMMING
(Common to CSE, IT)

Time: 3 Hours **Max. Marks: 75**

Note: This question paper contains two parts A and B.
Part A is compulsory which carries 25 marks. Answer all questions in Part A. Part B consists of 5 Units. Answer any one full question from each unit. Each question carries 10 marks and may have a, b as sub questions.

PART – A

(25 Marks)

1.a) State any four applications where python is more popular. [2]
b) List out the main differences between lists and tuples. [3]
c) What are the uses of File object? [2]
d) Give a brief description of several Built-in attributes related to File objects. [3]
e) Summarize the purpose of pipe and dot symbols used for pattern matching. [2]
f) Explain the basic functionality of match( ) function. [3]
g) What is the need of Tkinter module in python? [2]
h) How to create Label widget in Python? [3]
i) State the need of persistent storage. [2]
j) Discuss the SQL commands/statements used for creating, using and dropping a database. [3]

PART – B

(50 Marks)

2.a) How to declare and call functions in Python programs? Illustrate with an example script.
b) List and explain few most commonly used built-in types in python. [5+5]

OR

3. Summarize various operators, built-in functions and standard library modules that deals with Python's numeric type. [10]

4. Explain the following file built-in functions and method with clear syntax, description and illustration:
a) open( ) b) file( ) c) seek( ) d) tell( ) e)read( ) [10]

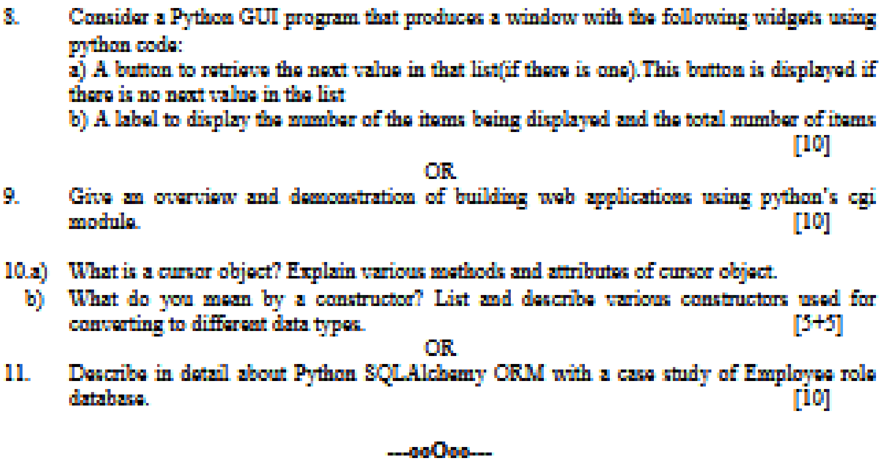
OR

5.a) How does try-except statement work? Demonstrate with an example python code.
b) Illustrate the concept of importing module attributes in python scripts. [5+5]

6. Examine how python supports regular expressions through the're' module with brief introduction and various built-in methods related to it. [10]

OR

7.a) What is the motivation behind parallelism and state how python achieves parallelism?
b) Explain briefly about thread and threading module objects in Python. [3+7]



8. Consider a Python GUI program that produces a window with the following widgets using python code:
a) A button to retrieve the next value in that list(if there is one).This button is displayed if there is no next value in the list
b) A label to display the number of the items being displayed and the total number of items [10]

OR

9. Give an overview and demonstration of building web applications using python's cgi module. [10]

10.a) What is a cursor object? Explain various methods and attributes of cursor object.
b) What do you mean by a constructor? List and describe various constructors used for converting to different data types. [5+5]

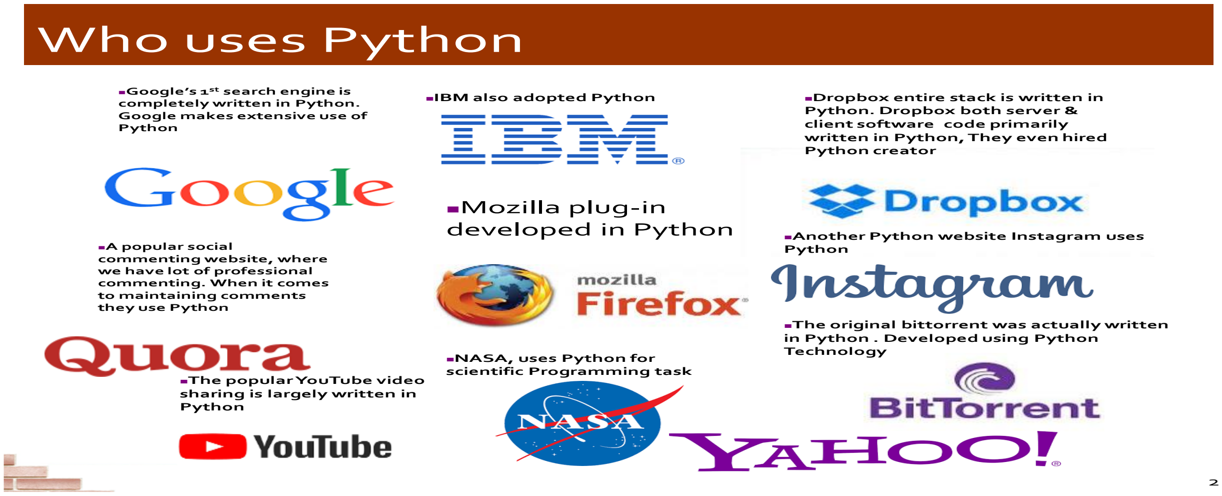
OR

11. Describe in detail about Python SQLAlchemy ORM with a case study of Employee role database. [10]

---ooOoo---

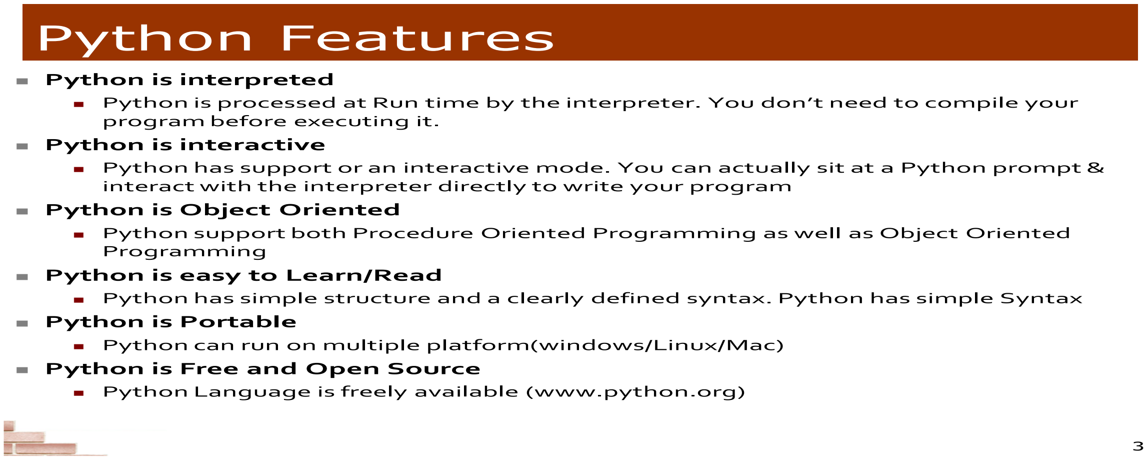
16. PowerPoint Presentation PPTs

Python Programming



Who uses Python

- Google's 1st search engine is completely written in Python. Google makes extensive use of Python
- IBM also adopted Python
- Dropbox entire stack is written in Python. Dropbox both server & client software code primarily written in Python, They even hired Python creator
- Mozilla plug-in developed in Python
- A popular social commenting website, where we have lot of professional commenting. When it comes to maintaining comments they use Python
- Another Python website Instagram uses Python
- The original bittorrent was actually written in Python . Developed using Python Technology
- NASA, uses Python for scientific Programming task
- The popular YouTube video sharing is largely written in Python

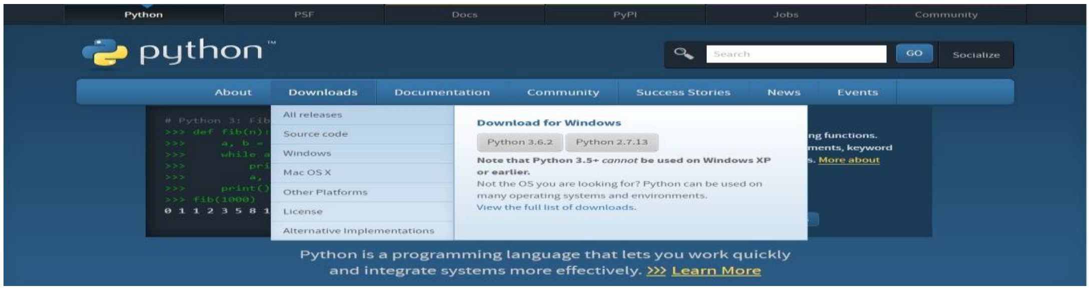


Python Features

- **Python is interpreted**
  - Python is processed at Run time by the interpreter. You don't need to compile your program before executing it.
- **Python is interactive**
  - Python has support or an interactive mode. You can actually sit at a Python prompt & interact with the interpreter directly to write your program
- **Python is Object Oriented**
  - Python support both Procedure Oriented Programming as well as Object Oriented Programming
- **Python is easy to Learn/Read**
  - Python has simple structure and a clearly defined syntax. Python has simple Syntax
- **Python is Portable**
  - Python can run on multiple platform(windows/Linux/Mac)
- **Python is Free and Open Source**
  - Python Language is freely available (www.python.org)

How can i get Python?

- Python for Window/Mac/Unix/Linux is available from www.python.org
  - From the above link download latest version of Python IDE and Install, Recent version is Python 3.6.2



```
# Python 3: Fib
>>> def fib(n):
>>>     a, b =
>>>     while a
>>>         pri
>>>         a,
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 1
```

Download for Windows

Note that Python 3.5+ *cannot* be used on Windows XP or earlier.

Not the OS you are looking for? Python can be used on many operating systems and environments.

View the full list of downloads.

ng functions.
ments, keyword
s. More about

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

First Python program

- **Print function in Python is function that outputs to your console window**
- At the prompt (>>>) type:

```
>>> print (“Hello, World!”)
>>> print "Message“
```

Output:

```
Hello, world!
Message
```

Variables

- Python uses Dynamic typing. That is, no need to Declare variables to be a specific type.
- At the >>> prompt, do the following:

```
x=5
type(x)

x=“this is text”
type(x)

x=5.0
type(x)
```

input

- input : Reads a number from user input.
  - You can assign (store) the result of input into a variable.
  - Example:

```
name=input("What is your name?")
age = input("How old are you? ")
print ("Your name is", name)
print ("Your age is", age)
```

Output:

```
What is your name? ACE
How old are you? 28
Your name is ACE
Your age is 28
```

- **Exercise:** Write a Python program to take the marks of 5 subject and display the same.

Operators in Python

Python language supports following type of operators.

- Arithmetic Operators
- Comparison Operators
- Logical (or Relational) Operators
- Assignment Operators

Python Arithmetic Operators:

- Many operators in Python look familiar (+, -, *, /, %) but few are new to you (//, **)

Operator	Description	Example
+	Addition - Adds values on either side of the operator	a + b will give 30
-	Subtraction - Subtracts right hand operand from left hand operand	a - b will give -10
*	Multiplication - Multiplies values on either side of the operator	a * b will give 200
/	Division - Divides left hand operand by right hand operand	b / a will give 2
%	Modulus - Divides left hand operand by right hand operand and returns remainder	b % a will give 0
**	Exponent - Performs exponential (power) calculation on operators	a**b will give 10 to the power 20
//	Floor Division - The division of operands where the result is the quotient in which the digits after the decimal point are removed.	9//2 is equal to 4 and 9.0//2.0 is equal to 4.0

Python Comparison Operators:

Operator	Description	Example
==	Checks if the value of two operands are equal or not, if yes then condition becomes true.	(a == b) is not true.
!=	Checks if the value of two operands are equal or not, if values are not equal then condition becomes true.	(a != b) is true.
<>	Checks if the value of two operands are equal or not, if values are not equal then condition becomes true.	(a <> b) is true. This is similar to != operator.
>	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	(a > b) is not true.
<	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	(a < b) is true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	(a >= b) is not true.
<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	(a <= b) is true.

Python Logical Operators

Operator	Description	Example
and	Called Logical AND operator. If both the operands are true then then condition becomes true.	(a and b) is true.
or	Called Logical OR Operator. If any of the two operands are non zero then then condition becomes true.	(a or b) is true.
not	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false.	not(a and b) is false.

Python Assignment Operators:

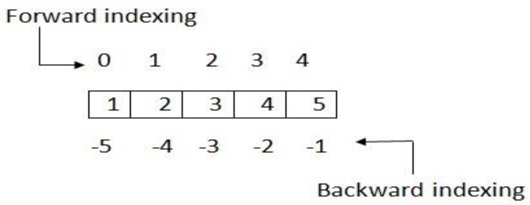
Operator	Description	Example
=	Simple assignment operator, Assigns values from right side operands to left side operand	c = a + b will assigne value of a + b into c
+=	Add AND assignment operator, It adds right operand to the left operand and assign the result to left operand	c += a is equivalent to c = c + a
-=	Subtract AND assignment operator, It subtracts right operand from the left operand and assign the result to left operand	c -= a is equivalent to c = c - a
*=	Multiply AND assignment operator, It multiplies right operand with the left operand and assign the result to left operand	c *= a is equivalent to c = c * a
/=	Divide AND assignment operator, It divides left operand with the right operand and assign the result to left operand	c /= a is equivalent to c = c / a
%=	Modulus AND assignment operator, It takes modulus using two operands and assign the result to left operand	c %= a is equivalent to c = c % a
**=	Exponent AND assignment operator, Performs exponential (power) calculation on operators and assign value to the left operand	c **= a is equivalent to c = c ** a
//=	Floor Division and assigns a value, Performs floor division on operators and assign value to the left operand	c //= a is equivalent to c = c // a

Sequences

- Sequence: an object that contains multiple items of data.
  - The items are stored in sequence one after another
- Python provides two types of sequences: **lists** and **tuples**
  - The difference between list and tuple is:
    - *Syntax* - Lists use [], tuples use ()
    - A list is mutable and a tuple is immutable(you cannot change the values in a tuple once you have created it).

Python List - Indexing

- Index: a number specifying the position of an element in a list
  - Enables access to individual element in list



- Forward indexing: Index of first element in the list is 0, second element is 1
- Backward indexing identifies the positions of element relative to the end of the list
  - The index -1 identifies the last element, -2 identifies the next to last element

Python – List Operations

- **Concatenate: Join two list together .The + operator can be used to concatenate two lists**
- **Example:**

```
>>> even_numbers=[2,4,6,8,10]
>>> odd_numbers=[1,3,5,7,9]
>>> numbers=even_numbers+odd_numbers
>>> numbers
[2, 4, 6, 8, 10, 1, 3, 5, 7, 9]
```

- **Replicating lists:** Replicating means repeating .
  - It can be performed by using '*' operator by a specific number of time.
- **Example:**

```
>>> even_numbers*2
[2, 4, 6, 8, 10, 2, 4, 6, 8, 10]
```

- **List Slicing**: A span of items that are taken from a sequence
  - **List slicing format : list _name [start_index : end_index]**
- **Example:**

```
>>> even_numbers
[2, 4, 6, 8, 10]
>>> even_numbers [ 0 : 2 ]
[2, 4]
```

Python List - functions & methods

- There are many Built-in functions and methods for Lists.
  - How Long is a List?
  - The len() function takes a list as a parameter and returns the number of *elements* in the list
  - Example: >>> My_List=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

```
>>> len(My_List)
10
```

- Minimum value from the list?
  - min(List_Name) function Returns the minimum value from the given list.
  - Example: >>> min(My_List)

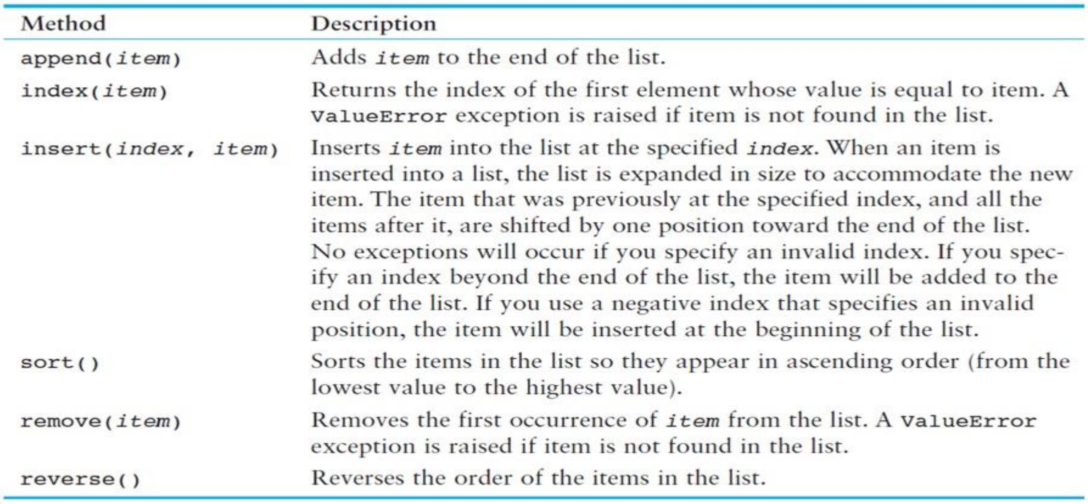
```
1
```

- Largest value from the given list ?
  - max(List_Name) function Returns the largest value from the given list.
  - Example: >>> max(My_List)

```
10
```

Python List - Methods

- There are many Built-in methods for Lists.



Method	Description
`append(item)`	Adds *item* to the end of the list.
`index(item)`	Returns the index of the first element whose value is equal to item. A `ValueError` exception is raised if item is not found in the list.
`insert(index, item)`	Inserts *item* into the list at the specified *index*. When an item is inserted into a list, the list is expanded in size to accommodate the new item. The item that was previously at the specified index, and all the items after it, are shifted by one position toward the end of the list. No exceptions will occur if you specify an invalid index. If you specify an index beyond the end of the list, the item will be added to the end of the list. If you use a negative index that specifies an invalid position, the item will be inserted at the beginning of the list.
`sort()`	Sorts the items in the list so they appear in ascending order (from the lowest value to the highest value).
`remove(item)`	Removes the first occurrence of *item* from the list. A `ValueError` exception is raised if item is not found in the list.
`reverse()`	Reverses the order of the items in the list.

Python Membership Operators:

- Python has membership operators, which test for membership in a sequence, such as strings, lists, or tuples.

Operator	Description
in	Evaluates to true if it finds a variable in the sequence and false otherwise.
not in	Evaluates to true if it does not finds a variable in the sequence and false otherwise.

```
>>> 'cs' in 'physics'
True
>>> ‘sleep' not in ‘CS 121'
True
 >>> 42 in [41,42,43]
 True
```

- **Exercise : Write a Python program to sum all the items in a list.**

Python - Tuples

- Tuples are another kind of sequence that function much like a list.
- Same as lists but
  - Immutable - once you create a tuple, you cannot change its contents.
  - Enclosed in parentheses.
  - A tuple with a single element ***must*** have a comma inside the parentheses (even though there is only one value):
    - **a = (11,)**

- **Creating a Tuple:** Creating a tuple is as simple as putting different comma-separated values between parentheses.
- Example:

```
>>>My_tuple=(23,'abc', 4.56, ' def')
>>> type(My_tuple)
<class 'tuple'>
```

Python Dictionary

- Dictionary is an unordered set of key and value pair, enclosed within curly braces.
- The key must be unique.
- The pair key and the value is separated by a colon(:).
- The pair i.e., key and value is known as item. Items are separated from each other by a comma(,).
- **Example:** # empty dictionary

```
>>>my_dict = { }

# dictionary with integer keys
>>>my_dict = {1: 'apple', 2: 'ball'}
```

17. Innovative Teaching Methods

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Approved by AICTE - New Delhi. Affiliated to JNTUH and Accredited by NAAC & NBA)

Kandlakoya (V), Medchal (M), Medchal - Malkajgiri (D)-501401

CMR
ENGINEERING COLLEGE
EXPLORE TO INVENT
NATIONAL BOARD of ACCREDITATION

Innovative Questions

Subject: Python Programming

Section- C

Date: 21/6/24

Q1. What are meta classes in Python and why are they important?

Q2. What is Django and how it's used by Python developers?

18. References (Websites/URLs/E-Resources)

i. https://nptel.ac.in/courses/106/106/106106212/

ii. https://www.w3schools.com/python/

iii. https://www.tutorialspoint.com/python/index.htm

iv. https://www.javatpoint.com/python-tutorial