BASIC PYTHON PROGRAMMING | **Course Code : 313011**

Programme Name/s	**: Automation and Robotics/ Digital Electronics/ Electrical Engineering/ Electronics & Tele-communication Engg./** **Electrical and Electronics Engineering/ Electrical Power System/ Electronics & Communication Engg./ Electronics Engineering/** **Instrumentation & Control/ Industrial Electronics/ Instrumentation/ Medical Laboratory Technology/** **Medical Electronics**
Programme Code	**: AO/ DE/ EE/ EJ/ EK/ EP/ ET/ EX/ IC/ IE/ IS/ ML/ MU**
Semester	**: Third / Fourth / Sixth**
Course Title	**: BASIC PYTHON PROGRAMMING**
Course Code	**: 313011**

I. RATIONALE

Electronics based industries needs to deal with creating circuits design, simulation, signal processing and control systems which can be developed using Python. This course deals with the basics of python to enhance the programming skills of diploma students. The course will enable students to write python programs as well as use different python libraries to solve given problems.

II. INDUSTRY / EMPLOYER EXPECTED OUTCOME

The aim of this course is to attain the following industry/employer expected outcome through various teaching learning experiences:
Develop programs using python to solve wide-reaching electronics engineering related problems.

III. COURSE LEVEL LEARNING OUTCOMES (COS)

Students will be able to achieve & demonstrate the following COs on completion of course based learning

- CO1 - Develop script to demonstrate use of basic building blocks of python.
- CO2 - Implement conditional and looping statements for given problem statement.
- CO3 - Perform operations on sequence structures in python.
- CO4 - Implement basics of object oriented programming concepts.
- CO5 - Create modules and packages for given purpose.

IV. TEACHING-LEARNING & ASSESSMENT SCHEME

Course Code	Course Title	Abbr	Course Category/s	CL	TL	LL	SLH	NLH	Credits	Paper Duration	FA-TH Max	SA-TH Max	Total Max	Total Min	FA-PR Max	FA-PR Min	SA-PR Max	SA-PR Min	SLA Max	SLA Min	Total Marks
313011	BASIC PYTHON PROGRAMMING	BPP	AEC	2	-	2	-	4	2	-	-	-	-	-	25	10	25@	10	-	-	50

Total IKS Hrs for Sem. : 0 Hrs

Abbreviations: CL- ClassRoom Learning , TL- Tutorial Learning, LL-Laboratory Learning, SLH-Self Learning Hours, NLH-Notional Learning Hours, FA - Formative Assessment, SA -Summative assessment, IKS - Indian Knowledge System, SLA - Self Learning Assessment

Legends: @ Internal Assessment, # External Assessment, *# On Line Examination , @$ Internal Online Examination

Note :

1. FA-TH represents average of two class tests of 30 marks each conducted during the semester.
2. If candidate is not securing minimum passing marks in FA-PR of any course then the candidate shall be declared as "Detained" in that semester.
3. If candidate is not securing minimum passing marks in SLA of any course then the candidate shall be

BASIC PYTHON PROGRAMMING | **Course Code : 313011**

declared as fail and will have to repeat and resubmit SLA work.
  4. Notional Learning hours for the semester are (CL+LL+TL+SL)hrs.* 15 Weeks
  5. 1 credit is equivalent to 30 Notional hrs.
  6. * Self learning hours shall not be reflected in the Time Table.
  7. * Self learning includes micro project / assignment / other activities.

V. THEORY LEARNING OUTCOMES AND ALIGNED COURSE CONTENT

Sr.No	Theory Learning Outcomes (TLO's)aligned to CO's.	Learning content mapped with Theory Learning Outcomes (TLO's) and CO's.	Suggested Learning Pedagogies.
1	TLO 1.1 Describe the given Keywords and Constants in Python. TLO 1.2 Use indentation, comments in the given program. TLO 1.3 Use different types of operators for writing expressions. TLO 1.4 Write python program using input output statements.	**Unit - I Basic Python's Constructs** 1.1 Introduction to Python- Python as scripting Language, Programming language Vs Scripting Language (C vs Python), Python's Technical Strength, Application in different domains 1.2 Python's building blocks- Identifiers, Keywords, Variables, Constants, Indentation, Comments in python 1.3 Python's Data Types – Numbers, Strings, List, Tuples, Dictionaries, Sets 1.4 Input and Output statements in python 1.5 Operators in Python- Operators as Arithmetic, Assignment, Unary Minus, Relational, Logical, Boolean, Bitwise, Membership, Identity, Operator precedence and Associativity	Presentations, Lecture Using Chalk-Board, Hands-on
2	TLO 2.1 Develop programs using Conditional Statements. TLO 2.2 Develop programs using Loop statements. TLO 2.3 Use statements to control the loops.	**Unit - II Control Statements in Python** 2.1 Types of Control Statements – Decision making statements, Looping statements 2.2 Decision Making Statements: - if, if….else, else-if ladder ,nested if and switch statement 2.3 Looping statement: - while loop, for loop, nested loop 2.4 Manipulating Loops- use of break, continue and pass statements	Lecture Using Chalk-Board, Demonstration, Hands-on, Flipped Classroom
3	TLO 3.1 Develop program to manipulate List for given purpose. TLO 3.2 Develop program to manipulate Tuples for given purpose. TLO 3.3 Develop program to manipulate Sets for given purpose. TLO 3.4 Develop program to manipulate Dictionaries for given purpose.	**Unit - III Data Structures in Python** 3.1 List- Defining List, Creating list, Accessing values from list, Updating the elements of a list, Concatenation of two lists, Repeating of Lists, Membership in list, Aliasing and cloning Lists, Methods to process Lists, Nested Lists 3.2 Tuples- Defining Tuple, Creating Tuples, Accessing the Tuple elements, Inserting elements in a Tuple, modifying elements of a Tuple, Deleting elements from a Tuple, Basic operations in Tuples, Functions to process Tuples, Nested Tuples 3.3 Sets- Defining Set, Creating a Set, Accessing elements from set, Add and update Set, Remove an elements from a Set, Built in functions with Set, Set methods to perform mathematical operations, other relevant set methods 3.4 Dictionaries- Defining Dictionary, Creating Dictionary, Accessing elements from Dictionary, Add and update Dictionary, Delete an element from a Dictionary, Built in functions of Dictionary, Methods to perform Dictionary	Demonstration, Lecture Using Chalk-Board, Hands-on
4	TLO 4.1 Use python built-in functions to perform tasks. TLO 4.2 Develop relevant user defined function for	**Unit - IV Functions with Basic OOP concepts** 4.1 Python Functions- Use of python built in functions (e.g. type/data conversion functions, math and string functions), User defined function- Function definition, function calling, function arguments and parameter passing, Return	Demonstration, Lecture Using Chalk-Board, Hands-on

BASIC PYTHON PROGRAMMING   **Course Code : 313011**

Sr.No	Theory Learning Outcomes (TLO's)aligned to CO's.	Learning content mapped with Theory Learning Outcomes (TLO's) and CO's.	Suggested Learning Pedagogies.
	the given purpose. TLO 4.3 Define classes to create and access objects with its methods and attributes.	statement, scope of variables (Global and Local Variables) 4.2 Basic OOP concepts- Introduction to object-oriented programming, Creating classes and objects, Constructors and Destructors in python, Data abstraction and Encapsulation	
5	TLO 5.1 Develop a python module in python for given purpose. TLO 5.2 Develop a python package for given purpose. TLO 5.3 Use NumPy for performing mathematical operations on arrays. TLO 5.4 Use matplotlib to create data visualization in python.	**Unit - V Modules and Packages in Python** 5.1 Modules- Writing modules, importing module, python built in modules (Numeric and mathematical module, Functional Programming Module) 5.2 Python packages- Introduction, Writing python packages, using standard packages (NumPy, matplotlib) and user defined package statements	Demonstration, Lecture Using Chalk-Board, Hands-on

VI. LABORATORY LEARNING OUTCOME AND ALIGNED PRACTICAL / TUTORIAL EXPERIENCES.

Practical / Tutorial / Laboratory Learning Outcome (LLO)	Sr No	Laboratory Experiment / Practical Titles / Tutorial Titles	Number of hrs.	Relevant COs
LLO 1.1 Install Python Integrated Development Environment.	1	a) Install and configure Python IDE. b) Write Python program to display message on screen.	2	CO1
LLO 2.1 Use operators in Python.	2	*a) Write simple Python program to calculate equivalent registers connected in series and parallel. Accept values of R1, R2 and R3 from the user. *b) Write simple Python program to calculate value of voltage by applying Ohm's law. Accept value of Current(I) and Resistance(R) from the user.	2	CO1
LLO 3.1 Implement two-way branching statement.	3	Write program to check whether entered frequency is radio frequency or audio frequency.	2	CO2
LLO 4.1 Implement multi-way branching statement.	4	*a)Write program to display various radio frequency bands using if..elseif ladder. *b)Write program to display resistor color code using switch statement.	2	CO2
LLO 5.1 Implement control loops for solving iterative problems.	5	*a. Write a simple Python program to demonstrate use of control loops: i) while ii) do while *b. Create a simple program, to demonstrate use of: for loop in Python (e.g.: various pattern building, printing multiplication table, checking palindrome number etc.)	2	CO2
LLO 6.1 Perform basic operations on the Lists.	6	*Write Python program to perform following operations on List: a) Create b) Access c) Update	2	CO3

BASIC PYTHON PROGRAMMING   **Course Code : 313011**

Practical / Tutorial / Laboratory Learning Outcome (LLO)	Sr No	Laboratory Experiment / Practical Titles / Tutorial Titles	Number of hrs.	Relevant COs
		d) Delete elements from list.		
LLO 7.1 Execute various tuple operations.	7	Develop Python program to perform following operations on Tuples: a) Create b) Access c) Update d) Delete Tuple elements	2	CO3
LLO 8.1 Implement various set operations.	8	Write Python program to perform following operations on Set: a) Create b) Access c) Update d) Delete Access Set elements	2	CO3
LLO 9.1 Execute various operations on Dictionaries.	9	*Create a program to perform following operations on Dictionaries in Python: a) Create b) Access c) Update d) Delete e) Looping through Dictionary	2	CO3
LLO 10.1 Use built-in mathematical functions and string functions in python.	10	a) *Create python program to demonstrate use of math built-in function. b) *Create python program to demonstrate use of string built-in function.	2	CO4
LLO 11.1 Create user defined functions in Python.	11	Write python programs to define function with arguments. a) Calculate factorial of a number b) Swapping of two variables	2	CO4
LLO 12.1 Implement function with default arguments.	12	Write programs to define function with default arguments.	2	CO4
LLO 13.1 Use built-in python mathematical modules.	13	*Create a program to demonstrate use of: Built-in module (e.g. numeric, mathematical functional and programming module) in Python.	2	CO5
LLO 14.1 Write user-defined module in python.	14	Write program to create a user-defined module (e.g.: building calculator) in python.	2	CO5
LLO 15.1 Use python built-in packages.	15	*Develop Python program to demonstrate use of NumPy package for creating, accessing and performing different array operations.	2	CO5
LLO 16.1 Implement user-defined packages in python.	16	Write program to demonstrate the use of user-defined packages in Python.	2	CO5

Note : Out of above suggestive LLOs -

- '*' Marked Practicals (LLOs) Are mandatory.
- Minimum 80% of above list of lab experiment are to be performed.
- Judicial mix of LLOs are to be performed to achieve desired outcomes.

VII. SUGGESTED MICRO PROJECT / ASSIGNMENT/ ACTIVITIES FOR SPECIFIC LEARNING / SKILLS DEVELOPMENT (SELF LEARNING)

Micro project

- Not Applicable

BASIC PYTHON PROGRAMMING | **Course Code : 313011**

Note :

- Above is just a suggestive list of microprojects and assignments; faculty must prepare their own bank of microprojects, assignments, and activities in a similar way.
- The faculty must allocate judicial mix of tasks, considering the weaknesses and / strengths of the student in acquiring the desired skills.
- If a microproject is assigned, it is expected to be completed as a group activity.
- SLA marks shall be awarded as per the continuous assessment record.
- For courses with no SLA component the list of suggestive microprojects / assignments/ activities are optional, faculty may encourage students to perform these tasks for enhanced learning experiences.
- If the course does not have associated SLA component, above suggestive listings is applicable to Tutorials and maybe considered for FA-PR evaluations.

VIII.   LABORATORY EQUIPMENT / INSTRUMENTS / TOOLS / SOFTWARE REQUIRED

Sr.No	Equipment Name with Broad Specifications	Relevant LLO Number
1	a) Computer System with all necessary peripherals and internet connectivity. b)Any relevant python IDE like IDLE/PyCharm/VSCode/Jupiter Notebook/Online Python Compiler.	All

IX.   SUGGESTED WEIGHTAGE TO LEARNING EFFORTS & ASSESSMENT PURPOSE (Specification Table)

Sr.No	Unit	Unit Title	Aligned COs	Learning Hours	R-Level	U-Level	A-Level	Total Marks
1	I	Basic Python's Constructs	CO1	4	0	0	0	0
2	II	Control Statements in Python	CO2	4	0	0	0	0
3	III	Data Structures in Python	CO3	10	0	0	0	0
4	IV	Functions with Basic OOP concepts	CO4	6	0	0	0	0
5	V	Modules and Packages in Python	CO5	6	0	0	0	0
		Grand Total		**30**	**0**	**0**	**0**	**0**

X.   ASSESSMENT METHODOLOGIES/TOOLS

Formative assessment (Assessment for Learning)

- Each practical will be assessed considering – 60% weightage to process and – 40% weightage to product.

Summative Assessment (Assessment of Learning)

- End semester summative assessment of 25 marks for laboratory learning.

XI.   SUGGESTED COS - POS MATRIX FORM

Course Outcomes (COs)	Programme Outcomes (POs)							Programme Specific Outcomes* (PSOs)		
	PO-1 Basic and Discipline Specific Knowledge	PO-2 Problem Analysis	PO-3 Design/ Development of Solutions	PO-4 Engineering Tools	PO-5 Engineering Practices for Society, Sustainability	PO-6 Project Management	PO-7 Life Long Learning	PSO-1	PSO-2	PSO-3

MSBTE Approval Dt. 02/07/2024   **Semester - 3 / 4 / 6, K Scheme**

BASIC PYTHON PROGRAMMING **Course Code : 313011**

					and Environment					
CO1	2							1		
CO2	2			1				2		
CO3	1	1	1	2				2		
CO4	1	2	2	2				2		
CO5	1	1	1	2				2		

Legends :- High:03, Medium:02,Low:01, No Mapping: -
*PSOs are to be formulated at institute level

XII. SUGGESTED LEARNING MATERIALS / BOOKS

Sr.No	Author	Title	Publisher with ISBN Number
1	Giancarlo Zaccone	Natural Computing with Python	BPB, ISBN:9789388511612
2	Martin C. Brown	Python: The Complete Reference	Tata McGraw Hill ISBN: 9789387572942
3	Yashwant Kanetkar	Let Us Python	BPB, ISBN: 978-9391392253
4	Kumar Naveen, Taneja Sheetal.	Python Programming: A modular approach	Pearson, ISBN: 978-9352861293
5	Mark Lutz and David Ascher	Learning Python	O'Reilly, ISBN: 978-1449355739
6	Paul Barry	Head First Python	O'Reilly, ISBN: 978-1449382674
7	John Guttag	Introduction to Computation and Programming Using Python	MIT Press, ISBN: 978-0262529624
8	David Beazley	Python Essential Reference	Addison-Wesley Professional, ISBN: 978-0672329784
9	Dr. R. Nageswara Rao	Core Python Programming	DREAMTECH PRESS, ISBN: 978-9386052308

XIII . LEARNING WEBSITES & PORTALS

Sr.No	Link / Portal	Description
1	https://www.programiz.com/python-programming	Python Programming
2	https://python-iitk.vlabs.ac.in/Introduction.html	Virtual Lab for Python Programming- Basic Constructs in Python
3	https://www.geeksforgeeks.org/python-programming-language/	Python Programming
4	https://intellipaat.com/academy/course/introduction-to-pytho n-programming-free-course/	Online Course-Python Programming
5	https://www.w3schools.com/python/	Python Programming
6	https://www.tutorialspoint.com/python/index.htm	Python Programming
7	https://www.python.org/	Python Programming
8	https://spoken-tutorial.org/tutorial-search/? search_foss=Pyt hon+3.4.3&search_language=English	Spoken Tutorial on Python Programming

Note :

- Teachers are requested to check the creative common license status/financial implications of the suggested online educational resources before use by the students