Lesson Plan No. 1	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: a. Getting familiar with the class b. Recognizes the difference between the Python and other languages like java, C++ c. Brief the history of Python and Python Versions.
Teaching Aids (if any)	a. ICT Usage
Teaching Development	1. **Introduction** (10 minutes) - Ask questions Which languages are used for backend and front end programming What is interpreter? What is the script language? Introducing myself to the class. - Brief History of Python Python Versions with the help of sides of ppt. - Discuss the importance of Python. 2. Introduction to Python programming and why python? (30 minutes) a) Discuss applications of python (such as game development, AI, Machine learning, Data sxience and desktop GUI [10] b) Discussion: programming languages - instructions to the computer; human-readable versus computer-readable, the difference between scripting languages (such as Python) versus compiled languages (such as C or C++) which would be used to create most of the programs the student might use on a day-to-day basis. [10 mins] c) Discussed the real examples where python is applied (5 minutes) d) Briefing Python platform and IDE(5 minutes)
Closure	5 minutes 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading https://www.edureka.co/blog/python-applications/ 3. Joy of computing (nptel) 4. Ebook https://drive.google.com/file/d/1POOjIxH0QbRtqQmLwFjCo7lrJDfPNset/view

Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 2	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: a. Gain the basic knowledge of Python Language b. Understand the features and uses of the language c. Articulate the Python application domains
Teaching Aids (if any)	a. ICT usage
Teaching Development	1. **Introduction** (5 minutes) • Ask questions What is interpreter? What is the script language? What is Python Language? • Encourage students to put up their ideas • Introduce the students with the basic terms related to the language 2. **Development** (30 minutes) a. Introduction to the main features of Python language • Explain each feature giving suitable examples b. Explain the uses of the language w.r.t. the other languages • Show suitable diagrams on slides • Give example to support the discussion c. Introduction to the application domains • Discuss the various application areas where Python is used now a days • Give real-time examples 3. Exercise (10 minutes) – • Have discussion to summarize the lecture • Ask Questions Related to Topic • Ask any student to recapitulate the topic done
Closure	1. Summarize the lesson learning outcomes and get affirmation from students on these. 2. Suggested reading and referring to the content https://www.python.org/about/gettingstarted/ Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. • Conduct a minute-paper activity Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 3	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: a. Gain knowledge on the basic syntax of Python Language. b. Understand the importance of indentation. c. Articulate different naming conventions of identifiers.
Teaching Aids (if any)	a. ICT usage
Teaching Development	1. **Introduction** (5 minutes) - Ask questions What is the functionality of interpreter? What IDE is used for python?? - Encourage students to put up their ideas - Introduce the students with the basic syntax related to the language 2. **Development** (30 minutes) a. Introduction to the basic syntax of Python language - Explain giving suitable examples. - Insert the missing part of the code below to output "Hello World". …….. ("Hello World") b. Explain the use and significance of indentation in python language - Give example to support the discussion c. Introduction to identifiers and keywords - Explain the need and naming convention of identifiers - Explaining the set of keywords used in Python. - Explaining the use of comments. 3. Exercise (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topic: Insert the missing indentation to make the code correct: if 5 > 2: print ("Five is greater than two!")

	Comments in Python are written with a special character,which one? ……it is a comment - Ask any student to recapitulate the topic done
Closure	1. Summarize the lesson learning outcomes and get affirmation from students on these. 2. Suggested reading and referring to the content https://www.python.org/about/gettingstarted/ Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. - Conduct a minute-paper activity Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 4	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: a. Gain knowledge on the concept of typecasting. b. Understand different types of data in Python. c. Articulate standard data types available in Python.
Teaching Aids (if any)	a. ICT Usage
Teaching Development	1. **Introduction** (5 minutes) - Ask questions What is the need of indentation in Python? What are the different naming conventions of identifiers? - Encourage students to put up their ideas on data types - Introduce the students with the need of type casting. 2. **Development** (30 minutes) a. Introduction to the setting and getting the data type - Explain the use of type() to get the type of the entered datagiving suitable examples. - Assigning a data type to a variable. - Example: The following code example would print the data type of x, what data type would that be? x = 5 print(type(x)) b. Explain different types of built in data types - Text - Numeric - Sequence - Mapping - Boolean - Binary c. Introduction to type conversion - Explain the need and syntax for type conversion - Example:Insert the correct syntax to convert x into a

	floating point number. x = 5 x = …..(x) - Explain the three numeric types(int, float, complex) available in python. 3. Exercise (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topic: - Insert the correct syntax to convert x into a complex number. x = 5 x = …….(x) - Insert the correct syntax to convert x into a integer. x = 5.5 x = ……(x) - Ask any student to recapitulate the topic done
Closure	1. Summarize the lesson learning outcomes and get affirmation from students on these. 2. Suggested reading and referring to the content https://www.python.org/about/gettingstarted/ Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. - Conduct a minute-paper activity Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 05	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: • Navigate PyCharm (create, save, run programs) • Create a simple program. Run it from the command line, and from the IDE(open the file and run it, all from within in the PyCharm
Teaching Aids (if any)	a. PPT presentation
Teaching Development	1. **Introduction** (10 minutes) - Brief review of prior lesson: Variables, numbers - What command are used for declaring a variable? - How can we check the type of variable? Introduce some of the Operators that are used to perform operations on variables and values. Give some examples. a) Download the latest version of Python and install (depending on the environment/school lab you may want to shortcut this process by pre downloading, making the installer accessible from the school network, and so on). b) Start up the Python Console, explain what the prompt is. Start up the Python Shell and compare. Try out basic print statements in both. c) Open a Python program and run using PyCharm or IDLE. d) Create a new window in PyCharm or IDLE, enter a simple program (hello world), save and run. Open the PyCharm again, try entering some basic commands like print. **Coding Activity (15)** • Enter Python code into PyCharm to demonstrate the above ideas.
Closure	5 minutes 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.

	2. Suggested Reading https://nptel.ac.in/courses/106/106/106106182/Joy of computing (nptel) -how to give instructions to pycharm
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 6	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: a. Gain knowledge on the concept of variables. b. Understand how to assign multiple values to variables. c. Articulate about local and global variables.
Teaching Aids (if any)	a. ICT Usage
Teaching Development	1. **Introduction** (5 minutes) - Ask questions What are the different naming conventions for identifiers? - Encourage students to put up their ideas on variables - Introduce the students to the difference between variable containers in python and other programming languages. 2. **Development** (30 minutes) a. Introduction to creating variables - Explain the use of quotes with string variables. - Explaining the case-sensitive nature of Python language. - Example: a = 4 A = "Sally" b. Explain the rules for python variables and categorizing different multi word variables - Camel case - Pascal case - Snake case c. Introduction to global keyword and global variables - Explain the need and syntax for type conversion - Example: 1. x = "awesome" def myfunc(): print("Python is " + x) myfunc() 2. def myfunc(): global x x = "fantastic" myfunc()

	print("Python is " + x) 3. Exercise (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topic: i. Create a variable named carname and assign the value Volvo to it. ii. Display the sum of 5 + 10, using two variables: x and y. ….. =….. y = 10 print(x…. y) - Ask any student to recapitulate the topic done
Closure	1. Summarize the lesson learning outcomes and get affirmation from students on these. 2. Suggested reading and referring to the content https://www.python.org/about/gettingstarted/ Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. - Conduct a minute-paper activity Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 7	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: a. Gain knowledge on the concept of strings. b. Understand how to perform slicing in strings. c. Articulate about the operations in strings.
Teaching Aids (if any)	a. Power-point presentation
Teaching Development	1. **Introduction** (5 minutes) - Ask questions What are the different types of data that are recognized in python? - Encourage students to put up their ideas on strings. - Introduce the students to the difference between numeric and string data types. 2. **Development** (30 minutes) a. Introduction to the string data type in python. - Explaining the use of single and double quotes with string variables. - Assigning multiline strings in Python b. Explaining the similarity between strings and arrays. - Indexing in a string (forward and backward) - Looping through a string - finding the length of the string. - Using “in” and “not in” keywords c. Introduction to slicing and modify operations in strings - Slicing from the start - Slicing to the end - Modify strings using lower(), upper(), replace() and split() 3. Exercise (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topic: i. Get the first character of the string txt. txt = "Hello World" x = …….

	ii. Return the string without any whitespace at the beginning or the end. txt = " Hello World " x =……. iii. Replace the character H with a J. txt = "Hello World" txt = txt…..(…,…..) - Ask any student to recapitulate the topic done
Closure	1. Summarize the lesson learning outcomes and get affirmation from students on these. 2. Suggested reading and referring to the content https://www.python.org/about/gettingstarted/ Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. - Conduct a minute-paper activity Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 08	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of branching in python • Implement if else statement in python
Teaching Aids (if any)	a. Power-point presentation
Teaching Development	1. **Introduction** (10 minutes) - Brief review of prior lesson: Operators in Python - If and Else if Syntax in Python ? - How and when we can use branching statements in Python ? Introduce if else statements in python that are used to create branching in python . **2. Development :** Give some examples. a) Program to print the largest of the three numbers. b) Program to check whether a person is eligible to vote or not c) Program to check whether a number is even or not.. 3. Exercise (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topic: **Coding Activity(15)** • Enter Python code into online compiler to demonstrate the above ideas.
Closure	5 minutes 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading https://nptel.ac.in/courses/106/106/106106182/Joy of computing (nptel)
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 09	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of looping • Implement looping using for and while loop
Teaching Aids (if any)	a. Power-point presentation
Teaching Development	1. **Introduction** (5 minutes) - Ask questions What are the different types of looping that are recognized in python? - Encourage students to put up their ideas using repeated task example. - Introduce the students to the difference between for and while loop. 2. **Development** (30 minutes) a. Introduction to the for loop in python. - Explaining the use of for loop with printing "Hello" multiple times . - Assigning various options in for loop b. Explaining the difference between for and while loop. c. Introduction to use of range function in for loop 3. Exercise (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topic: Count the total number of digits in a number Write a program to print multiplication table of a given number Calculate the sum of all numbers from 1 to a given number Ask any student to recapitulate the topic done **Coding Activity (15)** • Enter Python code into online compiler to demonstrate the above ideas.

Closure	5 minutes 1. Summarize the lesson learning outcomes and get affirmation from students on these. 2. Suggested reading and referring to the content https://www.python.org/about/gettingstarted/ Spend 5 minutes to wrap up and consolidate the leanings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 10	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of list in python • Implement the list in python
Teaching Aids (if any)	a. Power-point presentation
Teaching Development	1. **Introduction** (10 minutes) - Brief review of prior lesson: Looping in Python - Lists What they are and its Syntax in Python? - How and when we can use Lists in Python? Introduce Lists in python that are used to create structures in python. Give some examples. **2.Development** a) Create a simple program (Program to Find Even Numbers from a List), save and run b) Write a Python program to create a list with different data types c) Python program to right rotate the elements of a list **Coding Activity (15)** • Enter Python code into online compiler to demonstrate the above ideas.
Closure	5 minutes 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading https://nptel.ac.in/courses/106/106/106106182/Joy of computing (nptel)
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 11	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of functions • Implement looping using for and while loop
Teaching Aids (if any)	a. ICT Usage
Teaching Development	1. **Introduction** (5 minutes) - Ask questions functions in python? - Encourage students to put up their ideas using calling functions for performing different task example. A function definition a. Discuss Introduction to use of range function - Functions in Python - Python Lambda Expressions - Function Arguments - Python Recursion Function - Python Built-in Functions 2. **Development** (30 minutes) b. *# write Fibonacci series up to n ...* c. Specify a default value for one or more arguments. This creates a function that can be called with fewer arguments than it is defined to allow. 3. Exercise (10 minutes) – Use pre defined functions - range() - eval() - exec() **Coding Activity (15)** • Write a Python function to find the Max of three numbers Enter Python code into online compiler to demonstrate the above ideas.
Closure	5 minutes 1. Summarize the lesson learning outcomes and get affirmation from students on these. 2. Suggested reading and referring to the content

	The Fundamentals of Python: First Programs. Kenneth A. Lambert, Cengage Learning Page no. 160 to 171 https://www.w3schools.com/python/default.asp Spend 5 minutes to wrap up and consolidate the leanings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Model Institute of Engineering & Technology (Autonomous)

(Permanently Affiliated to the University of Jammu, Accredited by NAAC with "A" Grade)

Lesson Plan No. 12	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: a. Gain understanding about the concept of functions used in Python Language b. Articulate and understand creating, calling and passing values using function c. Understand the difference between parameters and arguments using examples
Teaching Aids (if any)	a. ICT usage
Teaching Development	1. **Introduction** (5 minutes) - Ask questions What are functions - Encourage students to put up their ideas - Introduce the student with the significance of using functions 2. **Development** (30 minutes) a. Introduction to creating, calling and passing values using functions - Creating function - Calling a function - Passing arguments b. Explain the difference between parameter and arguments - Suitable explanation - Showing various examples on slides explaining how values can passed, how to declare number of arguments, how to pass list as an argument c. Discussion on Pass Statement - Show few examples that explain usage of the statement - Discuss sample programs showing implementation of functions 3. Exercise (10 minutes) – - Give practice questions to students - Summarize the main points to remember - Ask questions/doubts if any
Closure	1. Summarize the lesson learning outcomes and get affirmation from students on these. 2. Suggested reading and referring to the content https://www.w3schools.com/python/python_functions.asp 3. Learn & practice https://www.programiz.com/python-programming/function

	Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. - Share a tutorial sheet with the students containing program list to be executed using functions Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 13	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of tuple in python • Implement the tuple in python
Teaching Aids (if any)	a. ICT Usage
Teaching Development	1. **Introduction** (10 minutes) - Brief review of prior lesson: List in Python - Lists What they are and its Syntax in Python ? - How and when we can use Tuples in Python ? Introduce Tuples in python that are used to create structures in python . Give some examples. **2.Development** a) Create a simple program (Program to Find Even Numbers From a Tuple , save and run b) Write a Python program to create a tuple with different data types c) Python program to left rotate the elements of an tuple **Coding Activity (15)** • Enter Python code into online compiler to demonstrate the above ideas.
Closure	5 minutes 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading https://nptel.ac.in/courses/106/106/106106182/Joy of computing (nptel) -how to give instructions to pycharm
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 14	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of sets in python. • Gain knowledge about the properties of sets. • Implement sets in python
Teaching Aids (if any)	a. ICT usage
Teaching Development	1. **Introduction** (5 minutes) - Ask questions - What are the tuples in Python? - Encourage students to put up their ideas on use of tuples. - Introduce the students to the use of sets in python 2. **Development** (30 minutes) a. Introduction to creating sets. - Explain the use of curly brackets with sets. - Explaining the set constructor. - Use of loops the accessing of elements in sets. b. Explain the properties of lists - Unindexed - Unordered - Unchangeable - Duplicates not allowed c. Introduction to different operations that can be performed on sets using built-in questions - Adding set items using add() and update() - Removing items using remove(), discard(), pop() and del 3. **Exercise** (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topic: i. Program to check if a set contains an element or not ii. Use the correct method to add multiple items (more_fruits) to the fruits set. fruits = {"apple", "banana", "cherry"}

	more_fruits = ["orange", "mango", "grapes"] - Ask any student to recapitulate the topic done
Closure	1. Summarize the lesson learning outcomes and get affirmation from students on these. 2. Suggested reading and referring to the content https://www.python.org/about/gettingstarted/ Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 15	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of dictionary in python • Implement the concept of dictionary in python
Teaching Aids (if any)	a. ICT Usage
Teaching Development	1. **Introduction** (10 minutes) - Brief review of prior lesson: Sets in Python - Lists What they are and its Syntax in Python ? - How and when we can use Dictionary in Python ? Introduce Dictionary in python that are used to create to store data values in key:value pairs in python . Give some examples. **2.Development** a) Python dictionary with keys having multiple inputs b) Python program to find the sum of all items in a dictionary c) Python program to find the size of a Dictionary **Coding Activity(15)** • Enter Python code into PyCharm to demonstrate the above ideas.
Closure	5 minutes 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading https://nptel.ac.in/courses/106/106/106106182/Joy of computing (nptel) -how to give instructions to pycharm
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 16	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: a. Gain understanding about the concept of other standard modules in python language b. Understand the implementation of each with the help of various examples
Teaching Aids (if any)	a. Power-point presentation b. Video Lecture
Teaching Development	1. **Introduction** (5 minutes) - Ask questions Recapitulate the old topic covered - What is the meaning of modules and its uses? - Encourage students to put up their ideas 2. **Development** (30 minutes) a. Introduction to the creation and usage of standard modules with examples - Math - Time - Sys b. Understand the significance of using these functions in Python - Suitable explanation - Showing various examples on slides 3. Exercise (10 minutes) – - Give practice questions to students - Summarize the main points to remember - Ask questions/doubts if any
Closure	1. Summarize the lesson learning outcomes and get affirmation from students on these. 2. Suggested reading and referring to the content 3. In Python, Modules are simply files with the ".py" extension containing Python code that can be imported inside another Python Program https://www.geeksforgeeks.org/python-math-module/ https://www.javatpoint.com/python-math-module Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.

	- MCQ based quiz sheet Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 17	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: a. Gain understanding about the concept of lambda module in python and its uses b. Understand the implementation of lambda function using examples
Teaching Aids (if any)	a. Power-point presentation b. Chalk & Talk c. Video Lecture
Teaching Development	1. **Introduction** (5 minutes) - Ask questions Recapitulate the old topic covered - What is the meaning of modules and its uses? - Encourage students to put up their ideas 2. **Development** (30 minutes) a. Introduction to the creation and usage of Lambda module with examples - Syntax - Example programs b. Understand the power of using lambda function used in Python - Suitable explanation - Showing various examples on slides 3. Exercise (10 minutes) – - Give practice questions to students - Summarize the main points to remember - Ask questions/doubts if any
Closure	1. Summarize the lesson learning outcomes and get affirmation from students on these. 2. Suggested reading and referring to the content https://www.javatpoint.com/python-lambda-functions 3. Learn & practice https://www.programiz.com/python-programming/anonymous-function#:~:text=Use%20of%20Lambda%20Function%20in%20python&text=In%20Python%2C%20we%20generally%20use,filter()%20%2C%20map()%20etc. https://www.w3schools.com/python/python_modules.asp Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. - MCQ based quiz sheet Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 18	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: a. Gain understanding about the concept of modules in python and its uses b. Understand the built-in modules used in python and the use of dir() function
Teaching Aids (if any)	a. Power-point presentation b. Chalk & Talk c. Video Lecture
Teaching Development	1. **Introduction** (5 minutes) - Ask questions Recapitulate the old topic covered - What is the meaning of modules? - Encourage students to put up their ideas - Introduce the student with the significance of using modules in python and its uses 2. **Development** (30 minutes) a. Introduction to creating and using modules with examples - Creating Module - Use a Module - Use variables in Module - Example discussion b. Understand about the built-in modules used in Python - Suitable explanation - Showing various examples on slides c. Using the dir() function - Definition and usage - Discuss sample programs showing implementation of dir() function d. Understand how to import from module - Suitable example explanation 3. Exercise (10 minutes) – - Give practice questions to students - Summarize the main points to remember - Ask questions/doubts if any
Closure	1. Summarize the lesson learning outcomes and get affirmation from students on these. 2. Suggested reading and referring to the content https://www.w3schools.com/python/python_modules.asp

	3. Learn & practice https://www.programiz.com/python-programming/modules Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. - MCQ based quiz sheet Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 19	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of file handling. b. Gain knowledge about the operations involved in file handling. c. Articulate the different modes of file handling in Python.
Teaching Aids (if any)	a. ICT usage
Teaching Development	1. **Introduction** (5 minutes) - Ask questions - What is the need of file handling? - Encourage students to put up their ideas on the use of file handling. - Introduce the students to the importance of file handling for nay web application. 2. **Development** (30 minutes) a. Introduction to the concept of file handling in Python b. Brief about basic syntax of different file handling operations - Open - Close - Read - Write c. Introduction to file object attributes. - File.closed - File.mode - File.name - File.softspace 3. **Exercise** (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topic: - Ask any student to **create a text file "intro.txt" in python and ask the user to write a single line of text by user input.** - **Create a text file "MyFile.txt" in python and ask the user to write separate 3 lines with three input statements from the user.**

Closure	1. Summarize the lesson learning outcomes and get affirmation from students on these. 2. Suggested reading and referring to the content https://www.python.org/about/gettingstarted/ https://www.tutorialspoint.com/python/python_files_io.htm Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 20	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of reading from files. b. Gain knowledge about different file methods. c. Articulate the handling exceptions while handling files in Python.
Teaching Aids (if any)	a. ICT usage
Teaching Development	1. **Introduction** (5 minutes) - Ask questions - What is the need of file handling? - Encourage students to put up their ideas on the use of file handling. - Introduce the students to the different modes in file handling. 2. **Development** (30 minutes) a. Introduction to the access modes to open a file in Python. - "r" - Read - "a" - Append - "w" - Write - "x" - Create - "t" - Text - "b" - Binary b. Modes while writing to an existing file - "a" - Append - "w" - Write c. Modes while creating a new file - "a" - Append - "w" - Write - "x" - Create 3. **Exercise** (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topic: - **Write a program to count a total number of lines and count the**

	total number of lines starting with 'A', 'B', and 'C'. (Consider the merge.txt file) - **Replace all spaces from text with – (dash).**
Closure	1. Summarize the lesson learning outcomes and get affirmation from students on these. 2. Suggested reading and referring to the content https://www.python.org/about/gettingstarted/ https://www.tutorialaicsip.com/cs-xii-pra/python-file-handling-programs/ Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 21	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: a. Understand how to delete a file. b. Gain knowledge about with statement in Python file handling. c. Articulate how to read files through loops.
Teaching Aids (if any)	a. ICT Usage
Teaching Development	1. **Introduction** (5 minutes) - Ask questions - What are the different operations that can be performed in python file handling? - Encourage students to put up their ideas on these methods and different modes used. - Introduce the students to the use of with statement. 2. **Development** (30 minutes) a. Introduction to the importance and the syntax of with statement. b. Brief about how to read file through a loop. c. Brief about how to read lines of a file. d. Different file methods in python e. Modifying file pointer positions - Offset - from 3. **Exercise** (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topic: - Ask any student to **Replace multiple spaces with single space in a text file.** - **Read the contents of file in reverse order**
Closure	1. Summarize the lesson learning outcomes and get affirmation from students on these. 2. Suggested reading and referring to the content https://www.python.org/about/gettingstarted/ https://www.tutorialspoint.com/python/python_files_io.htm Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 22	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: a. Understand how to perform directory management in Python. b. Gain knowledge about different modules available for directory management. c. Articulate how to compare directories.
Teaching Aids (if any)	a. ICT usage
Teaching Development	1. **Introduction** (5 minutes) - Ask questions - What are the different file management operations that can be performed in python? - Encourage students to put up their ideas on these methods and different modes used. - Introduce the students to the use of directory management. 2. **Development** (30 minutes) a. Introduction to the modules of directory management in Python. - os and os.path - filecmp - tempfile - shutil b. How to create a new directory? - **os.mkdir(name)** c. How to create a directory? - **os.rename()** d. Changing current working directory. - **os.chdir()** 3. **Exercise** (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topic: - Ask any student to **replace multiple spaces with single space in a text file.** - **Read the contents of file in reverse order**

Closure	1. Summarize the lesson learning outcomes and get affirmation from students on these. 2. Suggested reading and referring to the content https://www.python.org/about/gettingstarted/ https://www.geeksforgeeks.org/python-directory-management/ Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 23	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: a. articulate the concept of objects and class b. illustrate real world examples with OOPs in Python with examples
Teaching Aids (if any)	a. PPT presentation
Teaching Development	1. Introduction (5 minutes) - Ask questions What are classes What are the members of the class? What is the difference between instance variable and class variables? What is Instantiation in terms of OOP terminology? How we use class variables and what is its default access type - Introduce the concept of Access Modifier. The formal definition with the help of sides of ppt. - Talk about importance – Instance variable and class variables and methods. 2. Discuss an object (15) Explain its two characteristics with examples: attributes behavior Bundling data and functionality together in class. The __init__() Function and its use 3. Discussed the real examples for the above topic(15 minutes) Write the code in python to show the correct usage of public, private and protected member variables and methods. Ask students to share their screen to see how they are doing and rectify the program/s if they are getting any error while executing the program.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Access modifier topic discription and examples 3. Homework Write the programs to create base and derive classes and using all types of access modifiers, see how we can use them in our program 4. Spend 5 minutes to wrap up and consolidate the learning.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 24	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: a. Gain understanding about the concept Object Oriented Programming. b. Understand the importance of OOPs in Python. c. Articulate the difference between OOP and procedural programming.
Teaching Aids (if any)	a. Power-point presentation b. Chalk & Talk c. Video Lecture
Teaching Development	1. **Introduction** (5 minutes) - Ask questions Recapitulate the old topic covered - What is the meaning of OOPs and its advanatges? - Encourage students to put up their ideas 2. **Development** (30 minutes) a. Introduction to the features of OOPs - Class - Object - Method b. Understand the significance of the following principles in Python - Inheritance - Polymorphism - Data Abstraction - Encapsulation c. Explaining the difference between object oriented programming and procedural programming. 3. Exercise (10 minutes) – - Summarize the main points to remember - Ask questions/doubts if any
Closure	1. Summarize the lesson learning outcomes and get affirmation from students on these. 2. Suggested reading and referring to the content https://www.javatpoint.com/python-oops-concepts Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. - MCQ based quiz sheet Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 25	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: a. articulate the concept of Access Modifier in Python b. illustrate different Access Modifier in Python with examples
Teaching Aids (if any)	a. ICT Usage
Teaching Development	1. Introduction (5 minutes) - Ask questions What are classes What are the members of the class? What is the difference between instance variable and class variables? What is Instantiation in terms of OOP terminology? How we use class variables and what is its default access type - Introduce the concept of Access Modifier. The formal definition with the help of sides of ppt. - Talk about importance – Instance variable and class variables and methods. 2. Discuss three types of access modifiers: (25 minutes) • Public Access Modifier: Explain the Public variable and methods used in the class and in the program itself. Write program in Pycharm to illustrate public access modifier in a class. • Protected Access Modifier: Explain the Protected variable and methods used in the classes, derived classes and in the program itself. Write program in Pycharm to illustrate protected access modifier in a class. • Private Access Modifier: Explain the Private variable and methods used in the classes only. Write program in Pycharm to illustrate Private access modifier in a class. 3. Discussed the real examples for the above topic(10 minutes) Write the code in python to show the correct usage of public, private and protected member variables and methods. Ask students to share their screen to see how they are doing and rectify the program/s if they are getting any error while executing the program.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.

	2. Suggested Reading Access modifier topic discription and examples 3. Homework Write the programs to create base and derive classes and using all types of access modifiers, see how we can use them in our program 4. Spend 5 minutes to wrap up and consolidate the learning.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 26	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of classes and objects in Python. b. Gain knowledge about the self-argument in Python. c. Articulate the use of init() method in python.
Teaching Aids (if any)	a. ICT Usage
Teaching Development	1. **Introduction** (5 minutes) - Ask questions - What are the principles of OOP? - Encourage students to put up their ideas on use of sets. - Introduce the students to the use of classes and objects in python 2. **Development** (30 minutes) a. Introduction to classes and methods. - Creating a class - Creating a method b. Explain the properties of objects - State - Behaviour - identity c. Introduction to the self-parameter and init() method - Instantiating an object using constructors - Types of constructors - Use of destructors - Syntax of constructors and destructors 3. **Exercise** (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topic: - Ask any student to recapitulate the topic done
Closure	1. Summarize the lesson learning outcomes and get affirmation from students on these. 2. Suggested reading and referring to the content https://www.python.org/about/gettingstarted/ Spend 5 minutes to wrap up and consolidate the leanings.

Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 27	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of inheritance in Python. b. Gain knowledge about the types of inheritance in Python. c. Articulate the use of abstraction and overriding in python.
Teaching Aids (if any)	a. Power-point presentation
Teaching Development	1. **Introduction** (5 minutes) - Ask questions - What are classes and objects? - Encourage students to put up their ideas on use of sets. - Introduce the students to the use of classes and objects in python 2. **Development** (30 minutes) a. Introduction to concept and syntax of: - Inheritance - Parent class - Child class b. Explain the types of inheritance - Multiple - Multilevel c. Introduction to issubclass() and isinstance method - Method overriding - Data abstraction 3. **Exercise** (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topic: - Ask any student to recapitulate the topic done
Closure	1. Summarize the lesson learning outcomes and get affirmation from students on these. 2. Suggested reading and referring to the content https://www.python.org/about/gettingstarted/ https://www.geeksforgeeks.org/inheritance-in-python/?ref=lbp

	Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 28	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of polymorphism. b. Gain knowledge about the self-argument in Python. c. Articulate the use of init() method in python.
Teaching Aids (if any)	a. Power-point presentation
Teaching Development	1. **Introduction** (5 minutes) - Ask questions - What are the features of object oriented programming? - Encourage students to put up their ideas on need of polymorphism. - Introduce the students to the concept of polymorphism. 2. **Development** (30 minutes) a. Introduction to concept and syntax of polymorphism b. Explain the types of polymorphism - Operator - In-built functions - User- defined functions c. Introduction to relationship between polymorphism and inheritance. 3. **Exercise** (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topic: - Ask any student to recapitulate the topic done
Closure	1. Summarize the lesson learning outcomes and get affirmation from students on these. 2. Suggested reading and referring to the content https://www.python.org/about/gettingstarted/ https://www.geeksforgeeks.org/polymorphism-in-python/?ref=lbp Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 30	Course Name: Introduction to Programming with Python	Course No.: ESC-202

Objectives	At the end of the lesson the student shall be able to: a. Understand how to import the turtle module b. Draw simple shapes c. Basic understanding of what module in this class are there
Teaching Aids (if any)	a. PPT presentation b. Use of google meet
Teaching Development	1. Introduction (5 minutes) - Ask questions What GUI Introducing turtle library. - Discussion: o What is a module? o Builtin functions of Turtle class 2. Coding Activity (20 minutes) • Draw a line with the turtle. Discussion: a line is made up of pixels - what is a pixel? (possible additional activity: get a magnifying glass and see if students can see the edge of a pixel on the screen) • Turn the turtle 90 degrees and draw another line. Discussion: do the students know what degrees are? If not, use a clock to illustrate the major degrees (45, 90, 135, etc). Try turning the turtle 90 degrees right and 90 degrees left. Get the students to draw a zig-zag line. • Try drawing a square with different sizes. Reset and then try drawing a square on an angle. Try using the up and down commands to start and stop drawing. Generally free experimentation at this point. 3. Breakout room activity: (15 minutes) a. Create 6 groups and ask students to draw some unique pattern and one student from each group should share their screen and present their work
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Access modifier topic discription and examples 3. Homework Write the programs to create base and derive classes and using all types of access modifiers, see how we can use them in our program

	4. Spend 5 minutes to wrap up and consolidate the learning.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents