INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad -500 043

INFORMATION TECHNOLOGY

COURSE DESCRIPTOR

Course Title	**OBJECT ORIENTED PROGRAMMING THROUGH PYTHON**				
Course Code	AITB01				
Program	B.Tech				
Semester	THREE				
Course Type	Core				
Regulation	IARE - R18				
Course Structure	**Theory**			**Practical**	
	Lectures	**Tutorials**	**Credits**	**Laboratory**	**Credits**
	3		3	-	-
Course Coordinator	Ms. A Lakshmi, Assistant Professor				

I. COURSE OVERVIEW:

This course explains the fundamental ideas behind the object oriented approach to programming. Knowledge of python helps to create the latest innovations in programming. Like the successful computer languages that came before, python is the blend of the best elements of its rich heritage combined with the innovative concepts required by its unique environment. This course involves OOP concepts, python basics, inheritance, polymorphism, interfaces, packages, Exception handling. This course is presented to students by power point projections, course handouts, lecture notes, assignments, objective and subjective tests.

II. COURSE PRE-REQUISITES:

Level	Course Code	Semester	Prerequisites
B.Tech	ACSB01	II	Programming for problem solving

III. MARKS DISTRIBUTION:

Subject	SEE Examination	CIA Examination	Total Marks
Object Oriented Programming through Python	70 Marks	30 Marks	100

IV. DELIVERY / INSTRUCTIONAL METHODOLOGIES:

✔	PPT	✔	Chalk & Talk	✔	Assignments	✗	MOOCs
✔	Open Ended Experiments	✔	Seminars	✗	Mini Project	✔	Videos
✔	Others:						

V. EVALUATION METHODOLOGY:

The course will be evaluated for a total of 100 marks, with 30 marks for Continuous Internal Assessment (CIA) and 70 marks for Semester End Examination (SEE). Out of 30 marks allotted for CIA during the semester, marks are awarded by taking average of two CIA examinations or the marks scored in the make-up examination.

Semester End Examination (SEE): The SEE is conducted for 70 marks of 3 hours duration. The syllabus for the theory courses is divided into FIVE modules and each module carries equal weightage in terms of marks distribution. The question paper pattern is as follows. Two full questions with "either" or "choice" will be drawn from each module. Each question carries 14 marks. There could be a maximum of two sub divisions in a question.

The expected percentage of cognitive level of the questions is broadly based on the criteria given in Table: 1.

Table 1: The expected percentage of cognitive level of questions in SEE.

Percentage of Cognitive Level	Blooms Taxonomy Level
10 %	Remember
50 %	Understand
25 %	Apply
15 %	Analyze
0 %	Evaluate
0 %	Create

Continuous Internal Assessment (CIA):

CIA is conducted for a total of 30 marks (Table 2), with 20 marks for Continuous Internal Examination (CIE), 05 marks for Quiz and 05 marks for Alternative Assessment Tool (AAT).

Table 2: Assessment pattern for CIA

Component	Theory			Total Marks
Type of Assessment	CIE Exam	Quiz	AAT	
CIA Marks	20	05	05	30

Continuous Internal Examination (CIE):

Two CIE exams shall be conducted at the end of the 8th and 16th week of the semester respectively. The CIE exam is conducted for 20 marks of 2 hours duration consisting of five descriptive type questions out of which four questions have to be answered where, each question carries 5 marks. Marks are awarded by taking average of marks scored in two CIE exams.

Quiz - Online Examination

Two Quiz exams shall be online examination consisting of 25 multiple choice questions and are to be answered by choosing the correct answer from a given set of choices (commonly four). Such a question paper shall be useful in testing of knowledge, skills, application, analysis, evaluation and understanding of the students. Marks shall be awarded considering the average of two quiz examinations for every course.

Alternative Assessment Tool (AAT)

This AAT enables faculty to design own assessment patterns during the CIA. The AAT converts the classroom into an effective learning centre. The AAT may include tutorial hours/classes, seminars, assignments, term paper, open ended experiments, METE (Modeling and Experimental Tools in Engineering), five minutes video, MOOCs etc.

Table 3: Assessment pattern for AAT

5 Minutes Video	Assignment	Tech-talk	Seminar	Open Ended Experiment
30%	30%	30%	10%	---

VI. COURSE OBJECTIVES:

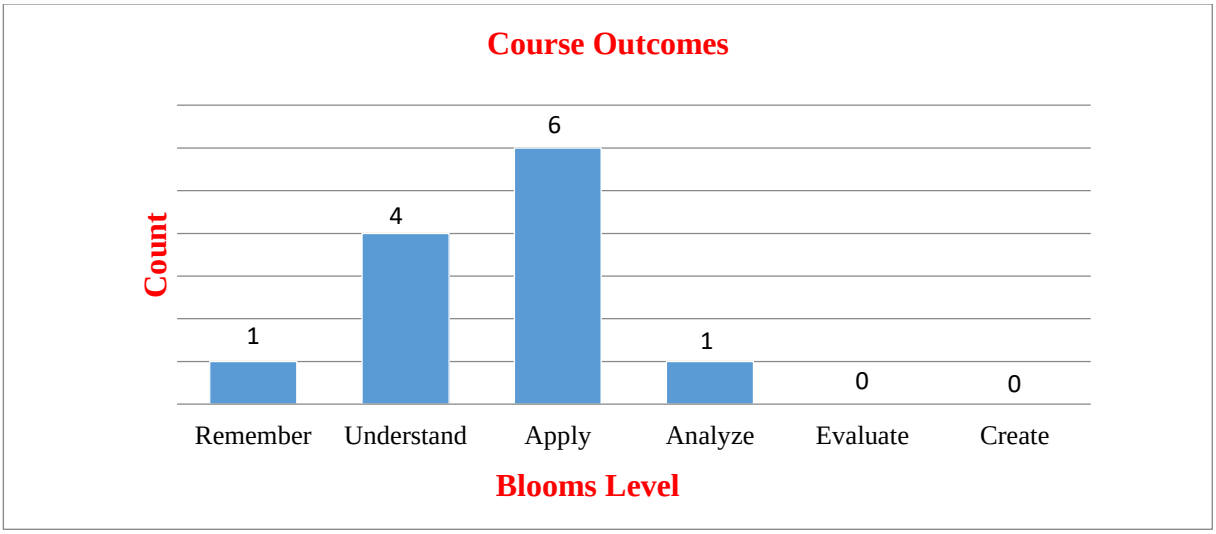
The students will try to learn:	
I	The fundamental concepts of object-oriented approach for solving real-time problems.
II	The basic and advanced constructs of Python programming for developing object oriented concepts.
III	The design concepts for developing user interface of real time applications.

VII. COURSE OUTCOMES:

After successful completion of the course, students will be able to::		
Course Outcomes		Knowledge Level (Bloom's Taxonomy)
CO 1	**Recall** the basic programming constructs in implementing in Python.	Remember
CO 2	**Identify** classes, objects, members of a class and relationship among them for real world entities.	Apply
CO 3	**Summarize** the object-oriented concepts such as Abstraction, Encapsulation, Inheritance and Polymorphism in real time context.	Understand

CO 4	**Demonstrate** abstraction feature with the help of python class properties	Understand
CO 5	**Make use** of polymorphism and inheritance concepts for achieving code reusability.	Apply
CO 6	**Apply** inbuilt strings for creating, performing basic operations and testing on text data.	Apply
CO 7	**Develop** user-defined functions for better modularity and a high degree of code reusability.	Apply
CO 8	**Explain** parameter-passing techniques while invoking recursive and non-recursive functions for solving problems.	Understand
CO 9	**Analyze** the Python exception mechanisms for handling errors and abnormal termination of program.	Analyze
CO 10	**Develop** user-defined exceptions for handling un-interrupted execution of specific programs.	Apply
CO 11	**Demonstrate** Python GUI tool kit for designing static user interfaces.	Understand
CO 12	**Make use** of widgets, containers and frames for creating user interface of web application.	Apply

COURSE KNOWLEDGE COMPETENCY LEVELS



VIII. HOW PROGRAM OUTCOMES ARE ASSESSED:

Program Outcomes		Strength	Proficiency Assessed by
PO1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineeringfundamentals, and an engineering specialization to the solution of complex engineering problems.	3	CIE/Quiz/AAT
PO2	**Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	2	CIE/Quiz/AAT

Program Outcomes		Strength	Proficiency Assessed by
PO 3	**Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with	3	Seminar/ Conferences
PO 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.	3	Assignments / Discussion

3 = High; 2 = Medium; 1 = Low

IX. HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED:

Program Specific Outcomes		Strength	Proficiency assessed by
PSO 3	Practical experience in shipping real world software, using industry standard tools and collaboration techniques will equip to secure and succeed in first job upon graduation in IT industry.	2	Group discussion / Short term courses

X. MAPPING OF EACH CO WITH PO(s), PSO(s):

Course Outcomes	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO 1	√														
CO 2	√	√													
CO 3	√														
CO 4	√														
CO 5	√	√													√
CO 6	√	√													√
CO 7	√	√	√												√
CO 8	√	√													
CO 9	√	√													√
CO 10	√	√	√												√
CO 11	√	√	√		√										√
CO 12	√	√	√		√										√

XI. JUSTIFICATIONS FOR CO-PO MAPPING:

Course Outcomes	POs / PSOs	Justification for mapping (Students will be able to)	No. of key competencies
CO 1	**PO 1**	Recall (knowledge) the basic programming constructs such as variables, operators, control statements and their importance and applicability (apply) in *implementing (complex) in python* by applying the principles of **mathematics and** basic programming **engineering fundamentals**.	2
CO 2	**PO 1**	Identify (knowledge) the classes, objects, members of a class and relationship among them in *solving (complex) engineering problems* by applying the principle so **mathematics and engineering fundamentals**.	2
	PO 2	Understand the given **problem statement** and **formulate** *(complex) specific engineering problems* related to classes, objects from the **information and data collection.**	3
CO 3	**PO 1**	Summarize (knowledge) abstraction, encapsulation, inheritance (apply) and polymorphism etc., in *solving (complex) object oriented concepts* by applying the principles of **mathematics and engineering fundamentals**.	2
CO 4	**PO 1**	Demonstrate (Knowledge)the abstraction features in *solving (complex) engineering problems* with the help of python class properties by applying the principles of **mathematics and engineering fundamentals**.	2
CO 5	**PO 1**	Make use of (Apply) the polymorphism and inheritance concepts in *solving (complex) engineering problems* to achieving the code reusability by applying the principles of **mathematics and engineering fundamentals** of object oriented programming.	2
	PO 2	Understand the given **problem statement** and **formulate** *(complex) specific engineering problems* related to code reusability from the **collection of data and information** in reaching substantiated conclusions by the **interpretation of** variations in the **results.**	4
	PSO 3	Make use of the polymorphism and inheritance concepts using **industry standard tools and collaboration technique** in the field of object-oriented programming.	2
CO 6	**PO 1**	Apply inbuilt string functions and testing methods of *(complex) engineering problems* by applying string operations on text data and their **integration and support with other engineering disciplines, mathematical principles.**	2
	PO 2	Understand the given **problem statement** and **formulate** *(complex) specific engineering problems* related to string operations and testing methods from the **information and data collection.**	3
	PSO 3	Apply inbuilt string functions and testing methods in real world software, using **industry standard tools and collaboration technique** in the field of Pyhton programming.	2
CO 7	**PO 1**	Develop (knowledge, understand and apply) the user defined functions for better modularity (*complex) engineering problems* by applying the principles of	2

		mathematics and engineering fundamentals.	
	PO 2	Understand the given **problem statement** and **formulate high** degree of code reusability *(complex) engineering problems* from the **information and data collection** in reaching substantiated conclusions by the **Interpretation of results.**	4
	PO 3	**Understand the user needs** of user-defined functions, **use creativity** of code reusability in applying the methods of model analyses **for innovative solutions, evaluate the outcomes** of the model analysis for handling better modularity **to achieve engineering objectives**.	5
	PSO 3	Develop the user-defined functions for better modularity in real world software, using **industry standard tools and collaboration technique** in the field of Python programming.	2
CO 8	**PO 1**	Explain (understand) the parameter passing techniques in *solving (complex) the functions of engineering problems* by applying the principles of **mathematics and engineering fundamentals** of python programming.	2
	PO 2	Understand the given **problem statement** and **formulate** recursive and non-recursive functions *(complex) engineering problems* from the **collection of data and information** in reaching substantiated conclusions by the **interpretation of results.**	4
CO 9	**PO 1**	Analyze the python exception mechanisms (knowledge) of solving the errors in *solving (complex) engineering problems* related to programs by applying the principles of and their **integration and support with other engineering disciplines, mathematical principles.**	2
	PO 2	Understand the given **problem statement** and **formulate** the handling errors *(complex) engineering problems* in a program from the **information and data collection.**	3
	PSO 3	Analyze the python exception mechanisms, using **industry standard tools and collaboration technique** in the field of object-oriented programming.	2
CO 10	**PO 1**	Develop (knowledge, understand and apply) the user defined exceptions for handling uninterrupted execution (*complex) engineering problems* by applying the principles of **mathematics and engineering fundamentals**.	2
	PO 2	Understand the given **problem statement** and **formulate** the (*complex) engineering problems* of un-interrupted execution of specific programs from the **collection of data and information, develop solutions** based on the user defined exceptions, **validate** the un-interrupted execution in reaching substantiated conclusions by the **Interpretation of results.**	6
	PO 3	**Understand the user needs** of user-defined exceptions, **use creativity** of exception handling in applying the methods of model analyses **for innovative solutions, evaluate the outcomes** of the model analysis for handling un-interrupted execution **to achieve engineering objectives**.	5
	PSO 3	Develop user-defined exceptions in real world software, using **industry standard tools and collaboration technique** in the field of object oriented programming.	2

CO 11	**PO 1**	Demonstrate (understand, apply) the python graphical user interface (GUI) tool kit for designing static user interfaces (*complex) engineering problems* by applying the principles of **mathematics and engineering fundamentals**.	2
	PO 2	Understand the given **problem statement** and **formulate** the (*complex) engineering problems* of designing static user interfaces from the **information and data collection, develop solutions** based on the designing static user interfaces, **validate** the python GUI tool kit in reaching substantiated conclusions by the **Interpretation of results.**	6
	PO 3	**Understand the user needs** of designing static user interfaces, **use creativity** of python GUI tool kit in applying the methods of model analyses **for innovative solutions**, **evaluate the outcomes** of the model analysis for developing the web applications **to achieve engineering objectives**.	5
	PO 5	Create the Python GUI tool kit for designing static user interfaces *(complex) Engineering activities* in **Computer software.**	1
	PSO 3	Demonstrate the python graphical user interface (GUI) tool kit in real world software, using **industry standard tools and collaboration technique** in the field of web application programming.	2
	PO 2	Understand the given **problem statement** and **formulate** the (*complex) engineering problems* of creating user interface of web application from the **collection of data and information, develop solutions** based on the widgets, containers and frames, **validate** the web application in reaching substantiated conclusions by the **Interpretation of results.**	6
	PO 3	**Understand the user needs** of creating user interfaces, **use creativity** of widgets, containers and frames in applying the methods **for innovative solutions, evaluate the outcomes** of the model analysis for developing the web applications **to achieve engineering objectives**.	5
	PO 5	Create the widgets, containers and frames for creating user interface of web application *(complex) Engineering activities* in **Computer software.**	1
	PSO 3	Make use of the widgets, containers and frames in shipping real world software, using **industry standard tools and collaboration technique** in the field of web application programming.	2

XII. TOTAL COUNT OF KEY COMPETENCIES FOR CO – (PO, PSO) MAPPING

Course Outcomes	**Program Outcomes / No. of Key Competencies Matched**												**PSO / No. of key competencies**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**	**3**
	3	**10**	**10**	**11**	**1**	**5**	**3**	**3**	**12**	**5**	**12**	**12**	**2**	**2**	**2**
CO 1	2														
CO 2	2	3													

CO 3	2														
CO 4	2														
CO 5	2	4													2
CO 6	2	3													2
CO 7	2	4	5												2
CO 8	2	4													
CO 9	2	3													2
CO 10	2	6	5												2
CO 11	2	6	5		1										2
CO 12	2	6	5		1										2

XIII. PERCENTAGE FOR KEY COMPETENCIES FOR CO-PO MAPPING:

Course Outcomes	**Program Outcomes / No. of key competencies**												**PSO / No. of key competencies**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**	**3**
	3	**10**	**10**	**11**	**1**	**5**	**3**	**3**	**12**	**5**	**12**	**12**	**6**	**1**	**2**
CO 1	66.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CO 2	66.7	30.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CO 3	66.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CO 4	66.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CO 5	66.7	40.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0
CO 6	66.7	30.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0
CO 7	66.7	40.0	50.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0
CO 8	66.7	40.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CO 9	66.7	30.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0
CO 10	66.7	60.0	50.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0
CO 11	66.7	60.0	50.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0
CO 12	66.7	60.0	50.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0

XIV. COURSE ARTICULATION MATRIX (CO-PO/PSO MAPPING)

COs and POs and COs and PSOs on the scale of 0 to 3, **0** being **no correlation**, **1** being the **low correlation**, **2** being **medium correlation** and **3** being **high correlation**.

0– **0**≤ ***C***≤ 5%–Nocorrelation;
1 – 5 <***C***≤ 40%– Low/ Slight;
2 − 40 % <***C***< 60% –Moderate.
3 – 60% ≤ ***C***< 100% – Substantial /High

Course Outcomes	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO 1	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 2	3	1	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 3	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 4	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 5	3	1	-	-	-	-	-	-	-	-	-	-	-	-	3
CO 6	3	1	-	-	-	-	-	-	-	-	-	-	-	-	3
CO 7	3	1	2	-	-	-	-	-	-	-	-	-	-	-	3
CO 8	3	1	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 9	3	1	-	-	-	-	-	-	-	-	-	-	-	-	3
CO 10	3	3	2	-	-	-	-	-	-	-	-	-	-	-	3
CO 11	3	3	2	-	3	-	-	-	-	-	-	-	-	-	3
CO 12	3	3	2	-	3	-	-	-	-	-	-	-	-	-	3
TOTAL	36	15	8		6										21
AVERAGE	**3.0**	**1.6**	**2.6**		**3.0**										**3.0**

XV. ASSESSMENT METHODOLOGY –DIRECT

CIE Exams	PO 1, PO 2, PO 3,PO 5	SEE Exams	PO 1,PO 2, PO 3, PO 5	Assignments	PO 3,PO 5	Seminars	PO 3,PO 5
Laboratory Practices	-	Student Viva	-	Mini Project	-	Certification	-
Term Paper	PO 5						

XVI. ASSESSMENT METHODOLOGIES–INDIRECT

✔	Early Semester Feedback	✔	End Semester OBE Feedback
✗	Assessment of Mini Projects by Experts		

XVII. SYLLABUS

MODULE-I	INTRODUCTION TO PYTHON AND OBJECT ORIENTED CONCEPTS
Introduction to Python: Features of Python, Data types, Operators, Input and output, Control Statements. Introduction to Object Oriented Concepts: Features of Object oriented programming system (OOPS) - Classes and Objects, Encapsulation, Abstraction, Inheritance, Polymorphism.	
MODULE-II	**PYTHON CLASSES AND OBJECTS**
Classes and Objects: Creating a class, The Self variable, Constructor, Types of Variable, Namespaces, Types of Methods, Inheritance and Polymorphism – Constructors in inheritance, The super() method, Types of inheritance, Polymorphism, Abstract classes and Interfaces.	
MODULE-III	**STRINGS AND FUNCTIONS**
Strings: Creating strings and basic operations on strings, String testing methods. Functions: Defining a function, Calling a function, Returning multiple values from a function, Functions are first class objects, Formal and actual arguments, Positional arguments, Recursive functions.	
MODULE-IV	**EXCEPTION HANDLING**
Exception: Errors in a Python program, Exceptions, Exception handling, Types of exceptions, The Except block, The assert statement, user-defined exceptions.	
MODULE-V	**GRAPHICAL USER INTERFACE**
GUI in Python: The Root window, Fonts and colors, Working with containers, Canvas, Frames, Widgets Button widget, Label Widget, Message widget, Text widget, Radio button Widget, Entry widget.	
Text Books:	
1. R Nageswara Rao, Core Python Programming, Dreamtech press, 2017 Edition. 2. Dusty Philips, Python 3 Object Oriented Programming, PACKT Publishing, 2nd Edition 2015.	
Reference Books:	
1. Michael H.Goldwasser, David Letscher, Object Oriented Programming in Python, Prentice Hall; 1st Edition, 2007.	

XVIII. COURSE PLAN:

The course plan is meant as a guideline. Probably there may be changes.

Lecture No	Topics to be Covered	CO	Reference
1	Describe the Features of Python, Data types.	CO 1	T1:1.2
2-3	Summarize the concept of Operators, Input and output, Control Statements.	CO 1	T1:4,5,6
4-5	Identify the features of Object Oriented Programming System (OOPS),	CO 2,CO3	T12.3
6-7	Use the concept of Classes and Objects, Encapsulation.	CO 2,CO 3	T1:12.4,12.5
8-9	Describe Abstraction, Inheritance, and Polymorphism.	CO 2,CO 3	T1:12.6-12.8

Lecture No	Topics to be Covered	CO	Reference
10-11	Determine Creating a class, The Self variable.	CO 4	T1:13.1,13.2
12-13	Understand types of variable, Namespaces.	CO 4	T1:13.4,13.5
14-15	Determine types of Methods, Inheritance and Polymorphism.	CO 4,CO 5	T1:13.6,14
16-18	Use Constructors in inheritance, the super() method.	CO 5	T1:14.1,14.3
19-20	Illustrate types of inheritance, Polymorphism, Abstract classes and Interfaces.	CO 5	T1:14.4,14.6
21-22	Understand Creating strings and basic operations on strings.	CO 6	T1:8.1
23	Analyze the concept of String testing methods.	CO 6	T1:8.17
24-25	Defining a function.	CO 7	T1:9.2
26-27	Illustrate Calling a function.	CO 7	T1:9.3
28	Illustrate Returning multiple values from a function.	CO 7	T1:9.5
29	Contrast the Usage of Functions is first class objects.	CO 8	T1:9.6
30	Contrast the Usage of Formal and actual arguments.	CO 8	T1:9.8
31	Define Positional arguments, Recursive functions.	CO 8	T1:9.9,9.16
32-34	Discuss the concept of Errors in a Python program.	CO 9	T1:16.1
35	Understand Exceptions, Exception handling.	CO 9	T1:16.2,16.3
36	Summarize the concept of types of exceptions.	CO 9	T1:16.4
37	Discuss the Except block, the assert statement.	CO 9	T1:16.5,16.6
38	Understand the concept of user-defined exceptions.	CO 10	T1:16.7
39	Knowledge about the Root window, Fonts and colors.	CO 11	T1:22.2,22.3
40-41	Apply Working with containers, Canvas.	CO 11,CO 12	T1:22.4,22.5
42	Understand Widgets, Button widget, Label Widget.	CO 12	T1:22.7
43	Implement Message widget, Text widget.	CO 12	T1:22.11
44-45	Illustrate Radio button Widget, Entry widget.	CO 12	T1:22.8

Prepared by:
Ms. A. Lakshmi, Assistant Professor **HOD, IT**