Code No: 137GA **R16**

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

B. Tech IV Year I Semester Examinations, July/August - 2023

PRINCIPLES OF PROGRAMMING LANGUAGES

(Computer Science and Engineering)

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

Note: i) Question paper consists of Part A, Part B.
ii) Part A is compulsory, which carries 25 marks. In Part A, Answer all questions.
iii) In Part B, Answer any one question from each unit. Each question carries 10 marks and may have a, b as sub questions.

PART – A

(25 Marks)

1.a)	What is meant by feature multiplicity?	[2]
b)	Give an attribute grammar for a simple assignment statement.	[3]
c)	What is l-value of a variable?	[2]
d)	Illustrate short-circuit evaluation of an expression.	[3]
e)	Define positional parameter.	[2]
f)	What is parameter profile of a subprogram?	[3]
g)	What is multiple inheritance?	[2]
h)	What are the various kinds of concurrency in a program?	[3]
i)	Give examples for logic programming languages.	[2]
j)	List the key concepts of scripting language.	[3]

PART – B

(50 Marks)

2. Discuss the areas of computer applications and their associated languages' features. [10]

OR

3. Give an overview of formal methods for describing syntax and semantics. [10]

4.a) How are scope and lifetime of a variable related?
b) Briefly describe the two approaches to defining type equivalence. [5+5]

OR

5.a) Explain the primary design issues for arithmetic expressions.
b) What is a mixed mode assignment? [6+4]

6.a) Compare the parameter passing mechanisms of Python with Java.
b) Can a subprogram be passed as a parameter? Justify your answer with relevant example. [5+5]

OR

7.a) Make a comparison of abstract data types of ADA and C++.
b) Explain the semantics of calls and returns in subprogram. [5+5]

8. Demonstrate the concept of dynamic binding in C++ and evaluate this against Smalltalk. [10]

OR

9. Explain synchronous message passing concept with example in Ada language. [10]

10. Appraise the importance of functional programming languages and the fundamental concepts associated with them. [10]

OR

11. With suitable examples, discuss procedural abstraction in Python modules. [10]

---ooOoo---