NARSIMHA REDDY ENGINEERING COLLEGE

(Autonomous)

Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad

Accredited by NBA & NAAC with A Grade

COURSE FILE

Program Name :Btech

Name of the Course : Principles of Programming Languages

Course Code : CS3112PE

Semester and Year : III - I

Faculty Name : G. sangeetha

S.NO	CONTENTS
1	Vision, Mission, COs, POs, PSOs, PEOs
2	Academic calendar
3	Syllabus
4	CO/PO mapping
5	Nominal Rolls of the Students
6	Time table
7	Lesson Plan
8	Unit wise Question Bank
9	Old Question Papers
10	Question Papers (CIA & SEE)
11	Tutorial sheets
12	Learning Methodologies: Experiential learning (Industrial visits, Internships, Mini Projects, Academic Projects, Guest Lectures, Student Workshops etc.,), Problem Solving methodologies (assignments, quiz, case study etc.) Note:1. At least TWO learning Methodologies to be included in your course 2. The above methodologies for illustration, you may add more

13	Subject notes/PPTs/self study material
14	Feedback on Curriculum Design and development
15	CO/PO attainment, analysis and Action taken report

1.Vision, Mission, COs, POs, PSOs, PEOs

- **Vision of the Department:**

To emerge as a center of excellence with international reputation by adapting the rapid advancements in the computer specialization fields.

- **Mission of the Department:**

1. To provide a strong theoretical and practical background in the area of computer science with an emphasize on software development
2. To inculcate Professional behavior, strong ethical values, leadership qualities, research capabilities and lifelong learning.
3. To educate students to be effective problem solvers, apply knowledge with social sensitivity for the betterment of the society and humanity as a whole.

- **COs:**

CO1	Evaluate to enhance and express the syntx and semantics of programming language
CO2	Evaluate the names expressions or key concepts in the implementation of programming language
CO3	Demonstrate and able to analyze the concepts of subprograms and ADT
CO4	Examine and illustrate the concepts of OOPS and concurrency levels
CO5	Evaluate lambda terms using Functional,Logic and Scripting language and their benefits

- **POs:**

1	**PO1. Engineering knowledge:** Apply the knowledge of basic sciences and fundamental engineering concepts in solving engineering problems.
2	**PO2. Problem analysis:** Identify and define engineering problems, conduct experiments and investigate to analyze and interpret data to arrive at substantial conclusions.
3	**PO3. Design/development of solutions:** Propose an appropriate solution for engineering problems complying with functional constraints such as economic, environmental, societal, ethical, safety and sustainability.

4	**PO4. Conduct investigations of complex problems**: Perform investigations, design and conduct experiments, analyze and interpret the results to provide valid conclusions.
5	**PO5. Modern tool usage**: Select or create and apply appropriate techniques and IT tools for the design & analysis of the systems.
6	**PO6. 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	**PO7. Environment and sustainability**: Demonstrate professional skills and contextual reasoning to assess environmental or societal issues for sustainable development.
8	**PO8. Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9	**PO9. Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multi-disciplinary situations.
10	**PO10. Communication**: Communicate effectively among engineering community, being able to comprehend and write effectively reports, presentation and give / receive clears instructions.
11	**PO11. Project management and finance**: Demonstrate and apply engineering & management principles in their own / team projects in multidisciplinary environment.
12	**PO12. Life-long learning**: Recognize the need for, and have the ability to engage in independent and lifelong learning.

PSOs:

1. PSO1: To provide effective and efficient real time solutions using acquired knowledge in various domains to crack problem using suitable mathematical analysis, data structure and suitable algorithm

2. PSO2: To develop environmental and sustainable engineering solution having global and societal context using modern IT tools.
3. PSO3: To exhibit professional and leadership skills with ethical values dealing diversified projects with excellent communication and documentation qualities.

- **PEOs:**

1. PEO-I: To provide students with a solid foundation in mathematics, engineering, basic science fundamentals required to solve computing problems and also to pursue higher studies and research.
2. PEO-IITo train students with good Computer Science and Engineering breadth so as to comprehend, analyze, design and create innovative computing products and solutions for real life problems.
3. PEO-III To inculcate in students professional and ethical attitude, communication skills, teamwork skills, multi-disciplinary approach and an ability to relate computer engineering issues with social awareness.

3.SYLLABUS

- **Course Objectives**
  - Introduce important paradigms of Programming Languages
  - To Provide conceptual understanding ofhigh-level language design and implementation
  - Topic include programming paradigms;syntax and semantics,data types, Expressions and statements;subprogramsand blocks;abstract data types,concurrency;functional and logic programming languages;and scripting languages.

- **Course Outcomes**

  - Acquire the skills for expressing syntax and semantics informal notation.
  - Identify and apply suitable programing paradigm for a given computing application.
  - Gain knowledge of and able to compare the features of various programming language.

MODULE-I

Preliminary Concepts: Reasons for studying concepts of programming languages, programming domains, language evaluation criteria, influences on language design, language categories, language design trade-offs, implementation methods, programming environments, Evolution of Major Programming Languages.

Syntax and Semantics: General problem of describing syntax, formal methods of describing syntax, attribute grammars, describing the meanings of programs

MODULE-II

Names, Bindings, and Scopes: Introduction, names, variables, concept of binding, scope, scope and lifetime, referencing environments, named constants

Data types: Introduction, primitive, character, string types, user defined ordinal types, array, associative arrays, record, tuple types, list types, union types, pointer and reference types, type checking, strong typing, type equivalence

Expressions and Statements: Arithmetic expressions, overloaded operators, type conversions, relational and boolean expressions, short- circuit evaluation, assignment statements, mixed-mode assignment

Control Structures – introduction, selection statements, iterative statements, unconditional branching, guarded commands.

MODULE-III

Subprograms: Fundamentals of subprograms, design issues for subprograms, local referencing environments, parameter passing methods, parameters that are subprograms, calling subprograms indirectly, overloaded subprograms, generic subprograms, design issues for functions, user defined overloaded operators, closures, co routines

Implementing subprograms: General semantics of calls and returns, implementing simple subprograms, implementing subprograms with stack-dynamic local variables, nested subprograms, blocks, implementing dynamic scoping

Abstract Data types: The concept of abstraction, introductions to data abstraction, design

issues, language examples, parameterized ADT, encapsulation constructs, naming encapsulations

MODULE-IV

Object Oriented Programming: D**esign** issues for OOP, OOP in Smalltalk, C++, Java, Ada 95, Ruby, Implementation of Object-Oriented constructs.

Concurrency: introduction, introduction to subprogram level concurrency, semaphores, monitors, message passing, Ada support for concurrency, Java threads, concurrency in functional languages, statement level concurrency.

Exception Handling and Event Handling: Introduction, exception handling in Ada, C++, Java, introduction to event handling, event handling with Java and C#.

MODULE-V

Functional Programming Languages: Introduction, mathematical functions, fundamentals of functional programming language, LISP, support for functional programming in primarily imperative languages, comparison of functional and imperative languages

Logic Programming Language: Introduction, an overview of logic programming, basic elements of prolog, deficiencies of prolog, applications of logic programming.

Scripting Language: Pragmatics, Key Concepts, Case Study: Python – Values and Types, Variables, Storage and Control, Bindings and Scope, Procedural Abstraction, Data Abstraction, Separate Compilation, Module Library. (Text Book 2)

TEXT BOOKS:

1. Concepts of Programming Languages, Robert .W. Sebesta 10th edition, Pearson Education.

2. Programming Language Design Concepts, D. A. Watt, Wiley India Edition.

REFERENCE BOOK:

1. Programming Languages, A.B. Tucker, R.E. Noonan, TMH.

2. Programming Languages, K. C. Louden and K A Lambert., 2nd edition, Thomson 2003.

.

4.List of COs (Action verbs as per Bloom's Taxonomy)

Course Code.CO No	Course Outcomes (CO's)
At the end of the course student will be able to	
C415.1	Classify, Compare, Contrast, Explain, Illustrate the use of formal description for a programming language and different programming paradigms
C415.2	Apply, build, Develop, Model, and Construct syntax and semantics to different programming languages
C415.3	Choose, Define, Find, What, Why the basic concepts of Variables, Storage, Binding and Scope rules of different languages
C415.4	Construct, Improve, Design, Solve, and Develop a Concurrency Control
C415.5	Compare, Evaluate, Explain, Justify, Determine the basics of Prolog and ML, Haskell, Python languages

Course Outcome (CO)-Program Outcome (PO) Matrix:

	PO 1	PO[2]	PO[3]	PO[4]	PO[5]	PO[6]	PO[7]	PO[8]	PO[9]	PO[1 0]	PO[1 1]	PO[1 2]
CO[1]	3	2	2	2	2	3	2	2	3	3	0	2
CO[2]	2	2	2	2	2	1	1	0	1	0	0	1
CO[3]	3	2	1	3	2	2	1	1	0	0	0	2
CO[4]	3	1	1	1	2	2	1	1	0	0	0	1
CO[5]	3	2	1	3	2	2	1	1	0	0	0	1
Avera ge	**2.8**	**1.8**	**1.4**	**2.2**	**2**	**2**	**1.2**	**1**	**0.8**	**0.6**	**0**	**1.4**

5.Mapping of course outcomes with PSO's

CO- PSO mapping

	PSO[1]	PSO[2]	PSO[3]
CO[1]	1	1	0
CO[2]	1	1	0
CO[3]	1	3	1
CO[4]	1	1	0
CO[5]	1	0	0
Average	**1**	**1.2**	**0.2**

6.INDIVIDUAL TIME TABLE

M.SOWMYA

III-1 CSE

Day & Time	**I**	**II**	**III**	**IV**	**12.40 - 01.20**	**V**	**VI**	**VII**
	09.10 – 10.10	**10.10 – 11.00**	**11.00 – 11.50**	**11.50 – 12.40**		**01.20 – 02.20**	**02.20 – 03.10**	**03.10 – 04.00**
MON	DS			CS		PP LAB		
TUE		CS		AIML			AIML	
WED		DS				PP LAB		
THU	AIML	PP LAB					DS	
FRI		AIML		CS			CS	
SAT	AIML	DS	CS				DS	

7.DETAILED LECTURE PLAN

S.No	**Tentative Date**	**Topics as per JNTUH Syllabus**	**Topic Actually Covered**	**Suggested Book**	**Method of Teaching**
					BB/PPT
	UNIT I PRELIMINARY CONCEPTS				
1	23-08-2022	Reasons for studying	Reasons for studying	T1,T2	BB
2	24-08-2022	concepts of programming languages	concepts of programming	T1,T2	BB

			languages		
3	25-08-2022	Programming domains	Programming domai ns	T2	BB
4	26-08-2022	Language Evaluation Criteria	Language Evaluation Criteria	T2	BB
5	29-08-2022	influences on Language design, Language categories	influences on Language design, Language categories	T2	BB
6	30-08-2022	Programming Paradigms – Imperative,	Programming Paradigms – Imperative,	T2	BB
7	30-08-2022	Object Oriented,	Object Oriented,	T2	BB
8	01-09-2022	functional Programming,	functional Programming,	T2	BB
9	02-09-2022	Logic Programming	Logic Programming	T2	BB
10	05-09-2022	Programming Language Implementation – Compilation and Virtual Machines,.	Programming Language Implementation – Compilation and Virtual Machines,.	T2	BB
11	06-09-2022	programming environments	programming enviro nments	T2	BB
12	08-09-2022	general Problem of describing Syntax and Semantics,	general Problem of describing Syntax and Semantics,	T2	BB
16	09-09-2022	attribute grammars	attribute grammars	T2	BB
17	12-09-2022	describing the meanings of programs	describing the meanings of		

			programs		
	UNIT - II **NAMES, BINDINGS AND SCOPES**				
18	13-09-2022	Introduction, names, variables	Introduction, names, variables	T2	BB
19	15-09-2022	concept of binding, scope	concept of binding, scope	T2	BB
20	16-09-2022	scope and lifetime	scope and lifetime	T2	BB
21	19-09-2022	referencing environments, named constants	referencing environments, named constants	T2	BB
22	20-09-2022	Introduction, primitive, character, user defined, array, associative, record, union, pointer and reference types	Introduction, primitive, character, user defined, array, associative, record, union, pointer and reference types	T2	BB
23	22-09-2022	design and implementation uses related to these types.	design and implementation uses related to these types.	T2	BB
24	23-09-2022	Names, Variable, concept of binding, type checking,	Names, Variable, concept of binding, type checking,	T2	BB
25	26-09-2022	strong typing, type compatibility,	strong typing, type compatibility,	T2	BB

26	27-09-2022	named constants, variable initialization	named constants, variable initialization	T2	BB
27	29-09-2022	Arithmetic relational and Boolean expressions,	Arithmetic relational and Boolean expressions,	T2	BB
28	30-09-2022	Short circuit evaluation mixed mode assignment, Assignment Statements	Short circuit evaluation mixed mode assignment, Assignment Statements	T2	BB
29	10-10-2022	Control Structures – Statement Level	Control Structures – Statement Level	T2	BB
30	11-10-2022	Iteration, Compound Statements, Selection	Iteration, Compound Statements, Selection	T2	BB
31	13-10-2022	Unconditional Statements, guarded commands.	Unconditional Statements, guarded commands.	T2	BB
	UNIT - III **SUBPROGRAMS AND BLOCKS**				
32	14-10-2022	Fundamentals of sub-programs	Fundamentals of sub-programs	T2	BB
33	17-10-2022	Scope of life time of variables, static and dynamic scope	Scope of life time of variables, static and dynamic scope	T2	BB
34	18-10-2022	Design issues of subprograms and	Design issues of subprograms and	T2	BB

		operations, local referencing environments	operations, local referencing environments		
35	20-10-2022	Parameterpassingmethodsoverloaded subprograms	Parameter passing methods overloaded subprograms	T2	BB
36	21-10-2022	generic sub-programs, overloaded	generic sub-programs, overloaded	T2	BB
37	31-10-2022	sub-program names,	sub-program names,	T2	BB
38	01-11-2022	parameters that are design issues for functions user defined	parameters that are design issues for functions user defined	T2	BB
39	03-11-2022	operators	Operators	T2	BB
40	03-11-2022	co routines	co routines	T2	BB
41	04-11-2022	General semantics of calls and returns	General semantics of calls and returns	T2	BB
42	07-11-2022	implementing simple subprograms,	implementing simple subprograms,	T2	BB
43	10-11-2022	implementing subprograms with stack-dynamic local variables	implementing subprograms with stack-dynamic local variables	T2	BB
44	11-11-2022	nested subprograms, blocks	nested subprograms, blocks	T2	BB
45	14-11-2022	implementing dynamic scoping	implementing dynamic scoping	T2	BB
46	15-11-2022	The concept of abstraction, introductions to data	The concept of abstraction,	T2	BB

		abstraction	introductions to data abstraction		
47	17-11-2022	design issues, language examples,	design issues, language examples,	T2	BB
48	18-11-2022	parameterized ADT, encapsulation constructs	parameterized ADT, encapsulation constructs	T2	BB
49	21-11-2022	naming encapsulations	naming encapsulations	T2	BB
50	22-11-2022	**Design** issues for OOP, OOP in Smalltalk, C++, Java	**Design** issues for OOP, OOP in Smalltalk, C++, Java	T2	BB
UNIT - IV **OBJECT ORIENTED PROGRAMMING**					
51	28-11-2022	Ada 95, Ruby, Implementation of Object-Oriented constructs	Ada 95, Ruby, Implementation of Object-Oriented constructs	T2	BB
52	29-11-2022	introduction, introduction to subprogram level concurrency	introduction, introduction to subprogram level concurrency	T2	BB
53	01-12-2022	semaphores, monitors, message passing,	semaphores, monitors, message passing,	T2	BB
54	02-12-2022	Ada support for concurrency	Ada support for	T2	BB

			concurrency		
55	05-12-2022	Java threads, concurrency in functional languages,	Java threads, concurrency in functional languages,	T2	BB
56	06-12-2022	statement level concurrency.	statement level concurrency.	T2	BB
57	08-12-2022	Introduction, exception handling in Ada	Introduction, exception handling in Ada	T2	BB
58	09-12-2022	C++, Java, introduction to event handling	C++, Java, introduction to event handling	T2	BB
59	12-12-2022	event handling with Java and C#.	event handling with Java and C#.	T2	BB
UNIT-V **FUNCTIONAL PROGRAMMING**					
60	12-12-2022	Introduction, fundamentals of FPL, LISP,	Introduction, fundamentals of FPL, LISP,	T1,T2	BB
61	15-12-2022	ML, Haskell,	ML, Haskell,	T1,T2	BB
62	15-12-2022	application of Functional Programming Languages and comparison of functional and imperative Languages	application of Functional Programming Languages and comparison of functional and	T1,T2	BB

			imperative Languages		
63	16-12-2022	**Scripting Language:** Pragmatics, Key Concepts	**Scripting Language:** Pragmatics, Key Concepts	T1,T2	BB
64	16-12-2022	Case Study: Python- Values and Types, Variables,	Case Study: Python- Values and Types, Variables,	T1,T2	BB
65	19-12-2022	Storage and Control, Bindings and Scope	Storage and Control, Bindings and Scope	T1,T2	BB
66	19-12-2022	Procedural Absraction, Separate Compilation, Module Library.	Procedural Absraction, Separate Compilation, Module Library.	T1,T2	BB
67	20-12-2022	Introduction, an overview of logic programming,	Introduction, an overview of logic programming,	T1,T2	BB
68	21-12-2022	basic elements of prolog,	basic elements of prolog,	T1,T2	BB
69	22-12-2022	deficiencies of prolog,	deficiencies of prolog,	T1,T2	BB
70	23-12-2022	applications of logic programming.	applications of logic programming.	T1,T2	BB

TEXT BOOKS:

- Concepts of Programming Languages Robert .W. Sebesta 8/e, Pearson Education, 2008.
- Programming Language Design Concepts, D. A. Watt, Wiley dreamtech, rp-2007.

REFERENCE BOOKS:

- Programming Languages, 2nd Edition, A. B. Tucker, R. E. Noonan, TMH.
- Programming Languages, K. C. Louden, 2nd Edition, Thomson, 2003.
- LISP Patric Henry Winston and Paul Horn Pearson Education.
- Programming in Prolog, W. F. Clocksin & C. S. Mellish, 5th Edition, Springer.
- Programming Python, M. Lutz, 3rd Edition, O'reilly, SPD, rp-2007.
- Core Python Programming, Chun, II Edition, Pearson Education, 2007.
- Guide to Programming with Python, Michel Dawson, Thomson, 2008

8.Assignment Questions

DEPARTMENT OF CSE

III-B.TECH I SEMESTER- ASSIGNMENT: I

SUBJECT:PPL

- **Describe** the steps involved in the language evaluation criteria. **[CO1, PO1]**

- **Explain** with an example how operator associatively can be incorporated in grammars. **[CO1,PO1,PO2]**

- The levels of acceptance of any language depend on the language description. **Comment** on this. **[CO2,PO1,PO2]**

- **Describe** in your own words, the concept of orthogonality in programming language design. **[CO2,PO1,PO2]**

- **Evaluate** the two approaches for supporting dynamic allocation and deal location for dynamic length strings. **[CO3, PO1]**

ACCREDITED BY NBA & NAAC WITH A-GRADE
NARSIMHA REDDY ENGINEERING COLLEGE
PERMANENTLY AFFILIATED TO JNTUH, HYDERABAD - APPROVED BY AICTE, NEW DELHI
AN ISO 9001 : 2008 CERTIFIED INSTITUTE
HELP LINE: 040 - 29705454

DEPARTMENT OF CSE

III-B.TECH I SEMESTER- ASSIGNMENT: II

SUBJECT: PPL

- Write a recursive Python function that recursively computes sum of elements in a list of lists. Sample Input: [1, 2, [3,4], [5,6]] Expected Result: 21**[CO3, PO1,PO2]**

- **Write** a python script to print the following pattern?**[CO4,PO1,PO2]**

```
* * * *
 * * *
  * *
   *
```

- **Describe** about a function called oops that explicitly raises a Index Error exception when called. Then write another function that calls oops inside a try/except statement to catch the error. What happens if you change oops to raise Key Error instead of Index Error? Where do the names Key Error and Index Error come from? **[CO4,PO1,PO2]**

- **Write** a Python program that creates a GUI with a textbox, Ok button and Quit button. On clicking Ok, the text entered in textbox is to be printed in Python shell; on clicking Quit, the program should terminate.**[CO5,PO1,PO2]**

- **Illustrate** the internal structure of a typical Python module with the suitable example. **[CO5,PO1,PO2].**

9.Sample assignment scripts

10.Unit-wise course material

11. Mid exam question papers

ACCREDITED BY NBA & NAAC WITH A-GRADE

NARSIMHA REDDY ENGINEERING COLLEGE

PERMANENTLY AFFILIATED TO JNTUH, HYDERABAD - APPROVED BY AICTE, NEW DELHI

AN ISO 9001 : 2008 CERTIFIED INSTITUTE

HELP LINE: 040 - 29705454

DEPARTMENT OF CSE

III-B.TECH I SEMESTER – I MID EXAMINATION

SET - A

SUBJECT: PPL DATE:

MAX. MARKS: 10 TIME:

ANSWER ANY TWO QUESTIONS 2*5=10M

1. a) Explain about the preconditions and post conditions of a given statement mean in axiomatic semantics. [2M][CO1PO2,,PO3]

b) What is type checking? Differentiate between static and dynamic type checking and give their relative advantages. [3M][CO2,PO2]

2. a) Describe the steps involved in the language evaluation criteria.[3M][CO1,PO2]

b) What is aliasing? What are the problems associated with it?[2M][CO2,PO2]

3. a) Discuss about language recognizers and language generators. [2M][CO1,PO2]

b) Explain the different parameter passing methods with an example.

[3M][CO3,PO2,PO3]

4. a) Define an array? Explain how to initialize an array? Explain the different types of arrays .[3M][CO2,PO2,PO3]

b) Explain about co routines.[2M][CO3,PO3]

DEPARTMENT OF CSE AND ALL EMERGING BRANCH

III-B.TECH I SEMESTER – II MID EXAMINATION

SET - A

SUBJECT: PPL **DATE:**

MAX. MARKS: 10 **TIME:**

ANSWER ANY TWO QUESTIONS **2*5=10M**

1 a) Distinguish between Competitive Synchronization and Cooperation synchronization. [2M][CO3,PO2]

b) Explain the following with respect to LISP: data types, structures and LISP interpreter.

[3M][CO4,PO2,PO3]

2 a)Explain how Binary semaphore provides solution to the competition synchronization problem with

example pseudo code. [3M] [CO4,PO2,PO3]

b) Explain the common characteristics of the scripting languages. [2M] [CO5,PO2,PO3]

3 a) Explain the design issues of an exception handling system.[2M][CO4,PO2,PO3]

b) Explain about different Abstract data types in various languages. [3M][CO3,PO2]

4a) Write in lisp for implementing Fibonacci and factorial of 'n' number using functions .[3M][CO4,P02]

b) What is the fundamental difference between a C++ class and an Ada package?[2M][CO5,PO2]

12.Sample mid answer script (Best, Average & Low)

13.Material collected from Internet/Websites/ PPT's

WEBSITES:

1.https://www.geeksforgeeks.org

2. https://en.wikipedia.org.

3.https://www.tutorialspoint.com.

14.University Questions / Question Bank

UNIT: 1

SHORT QUESTIONS

- Define Programming Language.
- Difference between sentence and sentential form.
- Define Syntax and Semantics
- Define Parse trees.
- Define Denotational semantics.
- Define Operational semantics.
- Differentiate compiler and interpreter.
- Distinguish simplicity and orthogonality
- Describe language recognizers.
- List out language categories

LONG QUESTIONS

- Explain in detail various phases in the process of compilation?
- Give an example of how aliasing deters reliability?
- Why it is useful for a programmer to have some background in language design?
- How can knowledge of programming language characteristics benefit the whole computing community?
- Difference between compilation and interpretation?
- What role does the symbol table play in a complier?
- In what fundamental way do operational semantics and denotational semantics differ?

UNIT: 2

SHORT QUESTIONS

- Distinguish static and dynamic scoping.
- Define associative arrays.
- Define guarded commands?
- Distinguish named type and structure type compatibility.
- List the merits of sub range types.
- Differentiate union and enumeration.
- Define data type.
- List the merits of type checking.
- Define user defined data type.
- Define widening and narrowing conversions?

LONG QUESTIONS

- Define the following?
    - Stack Dynamic
    - Explicit Heap Dynamic
    - Implicit Heap Dynamic
    - Static
- Difference between Static and Dynamic strings?
- How does operand evaluation order interact with functional side effects?
- What are design issues for character string types?
- Explain pointers, reference types, design issues of pointers, operations on pointers, pointer problems and implementation of pointer and reference types?
- Describe how the pointers used in C and C++ with examples?

UNIT: 3

SHORT QUESTIONS

- Define scope and lifetime of a variable.
- Explain subprograms in ADA
- Differentiate shallow and deep binding.
- Define local referencing environment.
- Define pass by value
- Define pass by reference
- List the design issues of functions
- Define static scope
- Define dynamic scope
- Difference between procedure and co routines

LONG QUESTIONS

- What are the modes, the conceptual models of transfer, the advantages and disadvantages of pass by value, pass by result, pass by value-result and pass by reference parameter passing methods?
- Define Shallow and deep binding for referencing environment of subprograms that have been passed as parameter?
- Write about Co routines?
- Explain the considerations for choosing parameter passing methods?
- What are the Generic characteristics of subprogram?
- Explain, how multi dimensional arrays can passed as arguments in Ada language?

UNIT: 4

SHORT QUESTIONS

- Define an exception
- Explain threads in C#
- Define concurrency.

- Define monitors.
- Define mutual exclusion.
- Write about message passing
- Define data abstraction.
- Define an abstract data type.
- Write the applications of logic programming languages
- Define deadlock.

LONG QUESTIONS

- Explain the difference Physical and logical concurrency?
- What are three possible levels of concurrency in programs? Explain?
- What is the use of Friend function in C++, Explain with example?
- Write short notes on C# threads?
- What are different states a task can be? Explain?
- Explain In detail Cooperation synchronization?
- Explain detailed compassion of Exception handling capabilities of C++ and those of Java?
- Explain the following with examples?
    - Exception handler
    - Disabling an Exception
    - Continuation
- How are Exceptions bound to handlers in C++?

UNIT: 5

SHORT QUESTIONS

- Give the meaning of lazy evaluation
- Define procedural abstraction.

- List few characteristics of Python language.
- Define functional language.
- Define imperative language.
- Give the meaning of scripting language.
- List few examples of scripting languages.
- List keywords of Python language.
- List data types of Python language.
- Define the term separate compilation in Python.

LONG QUESTIONS

1) Explain
   - Common Lisp
   - Haskell
   - ML

2) Discuss the applications of functional languages?

3) Explain the difference between Imperative and functional languages?

4) Describe the semantics of COND and LET?

5. **Explain** the characteristics of scripting languages.

6. **List** the ways in which ML is significantly different from scheme.

7. **Give** example of Python code to find the roots of quadratic equation

University Question Paper

R20

Code No: CS3112PE

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

B.Tech III Year I Semester Examinations, May/June - 2019 PRINCIPLES OF PROGRAMMING LANGUAGES

(Computer Science and Engineering)

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

Note: This question paper contains two parts A andB.
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, c as subquestions.

PART - A

(25 Marks)

1.a) Explain about theVirtual Machine. [2]

- What are the uses ofattributegrammar? [3]
- Explain about the problems inunconditionalbranching. [2]
- Explain about the enumerateddatatype. [3]
- What are the characteristicsofsubprograms? [2]
- Explainabout coroutines. [3]
- Why does Java not haveadestructors? [2]
- What are the applications oflogicprogramming? [3]

- Describe the scoping ruleinML. [2]
- Explain about the fundamentalsofFPL. [3]

PART - B

(50 Marks)

2.a) Explain about the preconditions and postconditions of a given statement mean in axiomaticsemantics.

b) Describe the steps involved in the languageevaluationcriteria. [5+5]

OR

3.a) Explain the different categories oflanguages.

b) Draw and explain the flow chart forcompilationprocess. [5+5]

4.a) Explain about the mixed-mode assignments that are used in Ada and JavaLanguages.

b) Explain about the type compatibility withanexample. [5+5]

OR

5.a) What is type checking? Differntiate between static and dynamic type checking and give their relativeadvantages.

b) Define an array? Explain how to initialize an array? Explain the different types of arrays.[5+5]

6.a) Describe about the static and dynamic scope of variables with anexample.

b) Define sub program. What are the distinct categoriesofsubprograms. [5+5]

OR

7.a) b)	Explain about the generic subprograms in Ada with an example. Explain about the semantic models of parameter passing.	[5+5]
8.a) b)	Explain about the concurrency in Ada 95. Explain the basic elements of prolog.	[5+5]

OR

9.a) Explain how to handle the exceptions inAda.

b) What are the design issues of abstractdatatypes. [5+5]

10.a) Explain about the internal representation of two LISPlists.

b) Describe the scoping rule in common LISPandHaskell. [5+5]

OR

11.a) Compare the functional programming languages with imperativelanguages.

b) Write a LISP function Fib(n) that computes nthFibonaccinumber. [5+5]

---ooOoo---

R20

Code No: CS3112PE

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITYHYDERABAD

B. Tech III Year I Semester Examinations, November/December - 2018 PRINCIPLES OF PROGRAMMING LANGUAGES

(Computer Science and Engineering)

Time:3hours Max. Marks:75

Note: This question paper contains two parts A andB.

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, c as subquestions.

PART - A

(25 Marks)

1.a) Defineaxiomaticsemantics. [2]

- Give an attribute grammar for simpleassignmentstatements. [3]
- What do you mean by precisionandrange? [2]
- What is aliasing? What are the problems associatedwithit? [3]
- Differentiate between actual andformalparameters. [2]
- What are the three general characteristicsofsubprograms? [3]
- Define abstractdatatype. [2]
- What is the purpose of aC++destructor? [3]
- How Haskell is differentfromML? [2]
- What is procedural abstraction?Giveexample. [3]

PART - B

(50 Marks)

2.a) Discuss various programming domains and their associatedlanguages.

b) Describe the basic concept ofdenotationalsemantics. [6+4]

OR

3.a) What are the potential benefits of studying programming languageconcepts?

b) Explain with examples how syntactic design choicesaffectreadability. [5+5]

4.a) What do you mean by binding? Give examples of some of the bindings and their bindingtimes.

b) Evaluate the two approaches for supporting dynamic allocation and deallocation for dynamiclengthstrings.

[6+4]

OR

5.a) Explain in detail various design issues of character stringtypes.

b) Whataredanglingpointersandlostheap-dynamicvariables?Howaretheycreated? [4+6]

6.a) How co-routines are different from conventionalsubprograms?

b) Explain type checking technique inparameterpassing. [5+5]

OR

7.a) What is a subprogram? Discuss the design issues of subprograms.

b) Write a detailed note on localreferencingenvironments.[5+5]

8.a) What are the various methods of exception handling? Discuss.

b) How message passing is implemented in ADA?Giveexamples. [5+5]

OR

9.a) Explain how information hiding is provided in an ADApackage.

b) Discuss about the basic elements of Prologwith examples. [5+5]

10.a) Explain the important functions of LISP.

b) Discuss the key concepts ofscriptinglanguages. [5+5]

OR

11.a) What are the three features of Haskell that makes very different fromschema?

b) What are the data types supported inPython?Discuss. [5+5]

15.Advanced Topics beyond syllabus

- PYTHON
- PHP

16.Course Attainment value for Internal marks

III-CSE

	Mid 1	Mid 2	Quiz 1	Quiz 2	Assign-1	Assign-2	Internal
CO 1	**3**		**3**		**3**		3
CO 2	**3**		**3**		**3**		3
CO 3	**2**	**3**	**3**	**3**	**3**	**3**	2.833333
CO 4		**3**		**3**		**3**	3
CO 5		**3**		**0**		**3**	2
Attainment							**2.77**

III-CSE

	Mid 1	Mid 2	Quiz 1	Quiz 2	Assign-1	Assign-2	Internal
CO 1	3		1		3		2.333333
CO 2	3		0		3		2
CO 3	3	3	0	0	3	3	2

CO 4		3		0		3	2
CO 5		3		3		3	3
Attainment							**2.27**

III CSE

	Mid 1	Mid 2	Quiz 1	Quiz 2	Assign-1	Assign-2	Internal
CO 1	**3**		**3**		**3**		3
CO 2	**3**		**3**		**3**		3
CO 3	**1**	**3**	**2**	**3**	**3**	**3**	2.5
CO 4		**3**		**3**		**3**	3
CO 5		**3**		**2**		**3**	2.666667
Attainment							**2.83**

17.Course Attainment value for External marks

III CSE-C

		Attainment
EXTERNAL	CO 1	3
	CO 2	3
	CO 3	3
	CO 4	3
	CO 5	3

III CSE-B AND ALL EMERGING BRANCH

		Attainment
EXTERNAL	CO 1	3
	CO 2	3
	CO 3	3
	CO 4	3
	CO 5	3

III =I CSE-A

		Attainment
EXTERNAL	CO 1	2
	CO 2	2

	CO 3	2
	CO 4	2
	CO 5	2

18.Final course Attainment

Name of the subject: PPL									**Yr/Sem:-III/I**	
Batch:	**Academic Year: 2021-2022**								**Branch: CSE-A**	
Course Attainment										
Final Direct Course Attainment									**Final Indirect Course Attainment Calculation**	
	Mid 1	Mid 2	Quiz 1	Quiz 2	Assign-1	Assign-2	Internal	University		
CO 1	**3**		**3**		**3**		3	3	**CO 1**	**2.89**
CO 2	**3**		**3**		**3**		3	3	**CO 2**	**2.84**
CO 3	**2**	**3**	**3**	**3**	**3**	**3**	2.833333	3	**CO 3**	**2.89**
CO 4		**3**		**3**		**3**	3	3	**CO 4**	**2.91**
CO 5		**3**		**0**		**3**	2	3	**CO 5**	**2.93**
Attainment							**2.77**	3.00	Final Indirect Course attainment	2.89
Weightage							25%	75%		
Direct Total Attainment							**0.69**	2.25		
final direct course attainment							2.94			
Weightage							80%		20%	
Total Attainment							2.35		0.58	
Course Attainment								**2.93**		

Name of the subject: PPL									**Yr/Sem:-III/I**	
Batch:	**Academic Year: 2021-2022**								**Branch: CSE-B**	
Course Attainment										
Final Direct Course Attainment									**Final Indirect Course Attainment Calculation**	
	Mid 1	Mid 2	Quiz 1	Quiz 2	Assign-1	Assign-2	Internal	University		
CO 1	**3**		**3**		**3**		3	2	**CO 1**	**2.79**
CO	**3**		**3**		**3**		3	2	**CO 2**	**2.8**

2										
CO 3	**3**	**3**	**3**	**3**	**3**	**3**	3	2	**CO 3**	**2.76**
CO 4		**3**		**3**		**3**	3	2	**CO 4**	**2.78**
CO 5		**3**		**3**		**3**	3	2	**CO 5**	**2.87**
Attainment							**3.00**	2.00	Final Indirect Course attainment	2.80
Weightage							25%	75%		
Direct Total Attainment							**0.75**	1.5		
final direct course attainment							2.25			
Weightage							80%		20%	
Total Attainment							1.80		0.56	
Course Attainment							**2.36**			

Name of the subject: PPL								**Yr/Sem:-III/I**	
Batch:	**Academic Year: 2021-2022**							**Branch: CSE-C**	
Course Attainment									
Final Direct Course Attainment								**Final Indirect Course Attainment Calculation**	
	Mid 1	Mid 2	Quiz 1	Quiz 2	Assign-1	Assign-2	Internal	University		
CO 1	**3**		**3**		**3**		3	**2**	**CO 1**	**2.68**
CO 2	**3**		**3**		**3**		3	**2**	**CO 2**	**2.68**
CO 3	**1**	**3**	**2**	**3**	**3**	**3**	2.5	**2**	**CO 3**	**2.57**
CO 4		**3**		**3**		**3**	3	**2**	**CO 4**	**2.89**
CO 5		**3**		**2**		**3**	2.666667	**2**	**CO 5**	**2.94**
Attainment							**2.83**	2.00	Final Indirect Course attainment	2.75
Weightage							25%	75%		
Direct Total Attainment							**0.71**	1.5		
final direct course attainment							2.21			
Weightage							80%		20%	
Total Attainment							1.77		0.55	
Course Attainment							**2.32**			

19.CO to PO Attainment

Course Code	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C415	2.36	1.52	1.18	1.86	1.69	1.69	1.01	0.84	o.67	0.51	0.17	1.118	0.84	1.01	0.17

20.Remarks and Recommendation