Most of the students were not able to answer the following questions:-

1) Example of pure functional programming paradigm?

Ans. Haskell

2) Example of Logic programming paradigm?

Ans. Prolog

3) Work of Lexical analysis in compilation?

Ans. To validate the lexems/words.

4) What is the work of Parse tree?

Ans. To check the syntax/grammar during compilation.

5) Concept of Binding?

Ans. Should say all type of relation between variable and its type.

6) Types of Type checking?

Ans. Static (int a = 10;) and Dynamic ($a = 10).

7) Strong typing?

Ans. When variable type is defined (int a = 10).

8) What is subprogram?

Ans. Functions and procedures.

9) What are the fundamentals of subprogram?

Ans. You should say all the characterstics and terms of a function/subprogram.

10) What are design issues of funtions?

Ans. You should say about precautions taken duriong creating a function (i.e, its type, number of value return or side effects etc.).

11) What is local referencing environments ?

Ans. Say about local variables and nested subprograms.

12) Wht is generic subprogram?

Ans. A subprogram in which a memory location is capable to store different types of variables.

example: template in C++.

13) What is concurrency ?

Ans. Its simultaneous execution of processes/subprogram/program/statements etc.

14) What is semaphore?

Ans. Its a ineger variable (int S), which acceses only by wait() or signal(). Used to prevent critical section problem.

15) What are the basic elements of prolog?

Ans. Term, fact , rules.

16) What are the applications of Logic programming?

Ans. Parsing, simulation etc.

Related posts:

1. Sequence Control & Expression | PPL
2. PPL:Named Constants
3. Parse Tree | PPL | Prof. Jayesh Umre
4. Basic elements of Prolog
5. Loops | PPL | Prof. Jayesh Umre
6. Subprograms Parameter passing methods | PPL | Prof. Jayesh Umre
7. Programming Paradigms | PPL | Prof. Jayesh Umre
8. Subprograms Introduction | PPL | Prof. Jayesh Umre
9. Phases of Compiler | PPL | Prof. Jayesh Umre
10. Parse Tree | PPL
11. Influences on Language design | PPL | Prof. Jayesh Umre
12. Fundamentals of Subprograms | PPL | Prof. Jayesh Umre
13. Programming Paradigm
14. Influences on Language Design
15. Language Evaluation Criteria

16. OOP in C++ | PPL
17. OOP in C# | PPL
18. OOP in Java | PPL
19. PPL: Abstraction & Encapsulation
20. PPL: Semaphores
21. PPL: Introduction to 4GL
22. PPL: Variable Initialization
23. PPL: Conditional Statements
24. PPL: Array
25. PPL: Strong Typing
26. PPL: Coroutines
27. PPL: Exception Handler in C++
28. PPL: OOP in PHP
29. PPL: Character Data Type
30. PPL: Exceptions
31. PPL: Heap based storage management
32. PPL: Primitive Data Type
33. PPL: Data types
34. Programming Environments | PPL
35. Virtual Machine | PPL
36. PPL: Local referencing environments
37. Generic Subprograms
38. Local referencing environments | PPL | Prof. Jayesh Umre
39. Generic Subprograms | PPL | Prof. Jayesh Umre
40. PPL: Java Threads
41. PPL: Loops
42. PPL: Exception Handling

43. PPL: C# Threads
44. Pointer & Reference Type | PPL
45. Scope and lifetime of variable
46. Design issues for functions
47. Parameter passing methods
48. Fundamentals of sub-programs
49. Subprograms
50. Design issues of subprogram
51. Garbage Collection
52. Issues in Language Translation
53. PPL Previous years solved papers
54. Type Checking | PPL | Prof. Jayesh Umre
55. PPL RGPV May 2018 solved paper discussion| Prof. Jayesh Umre
56. PPL RGPV June 2017 Solved paper | Prof. Jayesh Umre
57. Concurrency
58. Basic elements of Prolog
59. Introduction and overview of Logic programming
60. Application of Logic programming
61. PPL: Influences on Language Design
62. Language Evaluation Criteria PPL
63. PPL: Sequence Control & Expression
64. PPL: Programming Environments
65. PPL: Virtual Machine
66. PPL: Programming Paradigm
67. PPL: Pointer & Reference Type
68. try-catch block in C++