Overloaded sub-programs, also known as function overloading, is a feature in many programming languages that allows multiple functions or subprograms to have the same name but different parameter lists.

This means that you can define multiple subprograms with the same name but with different input parameters or return types.

When a program calls an overloaded subprogram, the compiler or interpreter determines the appropriate subprogram to invoke based on the arguments provided.

Here are some key points regarding overloaded sub-programs:

1. Function Name Reuse:

Overloaded subprograms enable you to reuse the same function name for different operations or variations of a task. This can lead to cleaner and more expressive code by using the same name for related functionalities.

2. Different Parameter Lists:

Overloaded subprograms must have different parameter lists to distinguish them. This can include differences in the number of parameters, the types of parameters, or both.

3. Compile-Time Resolution:

The resolution of which overloaded subprogram to invoke is typically determined at compile time based on the arguments passed in the function call. The compiler matches the arguments with the available overloaded versions and selects the most appropriate one.

4. Return Type Overloading:

In some programming languages, such as C++, the return type of a function can also be part of the overloading criteria. This means that you can have two functions with the same name and parameter list but different return types.

5. Polymorphism:

Overloaded subprograms contribute to polymorphism, a concept in object-oriented programming that allows objects of different types to be treated uniformly. Polymorphism allows you to write code that can handle different types of inputs using the same function name, improving code reusability and flexibility.

6. Code Readability:

Overloaded subprograms can enhance code readability by providing intuitive function names that reflect their purpose or behavior. For example, you can have multiple “calculate” functions with different parameter lists, such as “calculate(int x)” and “calculate(int x, int y)”.

Related Posts:

1. Relationship among entities
2. Introduction of IOT
3. Marketing Managment RGPV Diploma Paper Solved
4. Value of function in programming
5. Hardware components and device solved paper RGPV Diploma
6. USE CASE for MCQ application
7. OS Interview Q & A | Part 01 | Prof. Jayesh Umre
8. Compilation

9. OOPs in C# | PPL | Prof. Jayesh Umre
10. Static and Dynamic scope
11. Type Checking
12. Testing Levels | Software engineering | SEPM | Prof. Jayesh Umre
13. Static and Dynamic Analysis | Software Engineering| SEPM| Prof. Jayesh Umre
14. Code Inspection | Software engineering | SEPM | Prof. Jayesh Umre
15. Code Inspection
16. Characterstics of IOT
17. IOT Internet of Things
18. Monitors
19. Static and Stack-Based Storage management
20. Message passing
21. Exception handler in Java
22. Exception Propagation
23. Concept of Binding
24. Data mining and Data Warehousing
25. Introduction to Concurrency Control
26. Introduction to Transaction
27. Introduction to Data Models
28. Coaxial Cable
29. DHCP
30. DNS
31. Introduction to SNMP
32. Introduciton to SMTP
33. Introduction to NFS
34. Introduction to Telnet
35. Introduction to FTP

36. Internet Intranet Extranet
37. UGC NET Notes
38. Computer Terminologies
39. UGC NET Paper 1 December 2012
40. UGC Net paper 1 June 2011
41. closure properties of regular languages
42. Functional programming languages
43. Virtualization fundamental concept of compute
44. Dia software for UML, ER, Flow Chart etc
45. DAVV MBA: Business Communication
46. Mirroring and Striping
47. RGPV Solved Papers
48. CD#08 | Semantic analysis phase of compiler in Hindi video | Semantic tree | Symbol table | int to real
49. COA#27 | Explain the Memory Hierarchy in short. | COA previoys years in Hindi video
50. Infix to Postfix expression
51. Array implementation of Stack
52. Stack Data Structure
53. DBMS#03 | DBMS System Architecture in Hindi video
54. Java program method overloaing
55. Java program use of String
56. DS#33 | 2 Dimensional Array | Data Structure in Hindi video
57. SE#10 | Function point (FP) project size estimation metric in Hindi video
58. ADA#02 | Define Algorithm. Discuss how to analyse Algorithm | ADA previous years in Hindi video
59. Principles of Programming Languages
60. Discrete Structures

61. Machine Learning
62. R Programming Video Lectures
63. Internet of Things (IOT)
64. Digital Circuits
65. Number Systems
66. Computer Organization and Architecture Video Lectures
67. UGC NET
68. There are five bags each containing identical sets of ten distinct chocolates. One chocolate is picked from each bag.The probability that at least two chocolates are identical is ______
69. C Programming Questions
70. What is Software ? What is the difference between a software process and a software product ?
71. Difference between scopus and sci/scie journal
72. Human Process Interventions: Individual and Group Level & Organization Level Topics Covered: Coaching, training and development, conflict resolution process process consultation, third-party interventions, and team building.
73. Leading and Managing Change & Emerging Trends in OD
74. Designing and Evaluating Organization Development Interventions
75. Tutorial
76. Data Dictionary and Dynamic Performance Views
77. Anna University Notes | Big Data Analytics
78. What is Map Reduce programming model? Explain.
79. Features of Web 2.0
80. Describe in brief the different sources of water.
81. RGPV BEEE
82. Define data structure. Describe about its need and types.Why do we need a data type

?

83. Interview Tips
84. Find output of C programs Questions with Answers Set 01