

## SEQUENCE CONTROL AND EXPRESSION

### Viva Voce on Sequence control & expression

Q1. Describe static function with its usage?

Ans) A function, which has a function definition prefixed with a static keyword is defined as a static function. The static function should call within the same source code.

Q2. Describe Wild Pointers in C?

Ans) Uninitialized pointers in the C code are known as Wild Pointers. These are a point to some arbitrary memory location and can cause bad program behaviour or program crash.

Q3. What is the difference between ++a and a++?

Ans) '++a' is called prefixed increment and the increment will happen first on a variable. 'a++' is called postfix increment and the increment happens after the value of a variable used for the operations.

Q4. Describe the difference between = and == symbols in C programming?

Ans) '=' is the comparison operator which is use to compare the value or expression on the left-hand side with the value or expression on the right-hand side.

'=' is the assignment operator which is use to assign the value of the right-hand side to the variable on the left-hand side.

Q5. What is the explanation for prototype function in C?

Prototype function is a declaration of a function with the following information to the compiler.

- Name of the function.
- The return type of the function.
- Parameters list of the function.

Q6. Describe the header file and its usage in C programming?

Ans) The file contains the definitions and prototypes of the functions being used in the program are called a header file. It is also known as a library file.

Q7. What is a nested loop?

Ans) A loop running within another loop is referred as a nested loop.

Q8. What is a pointer on a pointer in C programming language?

Ans) A pointer variable that contains the address of another pointer variable.

Q9. What are the valid places to have keyword "Break"?

Ans) Only in Looping or switch statements.

### MCQs on Sequence control on expressions

Q1. The break statement is used to exit from

- a. DO loop
- b. FOR loop
- c. all of above

Q2. In which statements, does a CONTINUE statement cause the control to go directly to the test condition and then continue the looping process?

- a. FOR and WHILE
- b. WHILE and IF-ELSE
- c. While and DO-WHILE

Q3. The advantage of a SWITCH statement over an ELSE-IF statement

- a. A default condition can be used in the SWITCH
- b. The SWITCH is easier to understand<sup>1</sup>
- c. Several different statements can be executed in a SWITCH

Q4. The traditional way to create an infinite loop in C is

- a. FOR ( ; ; )

b. IF (=) BREAK;

c. WHILE ()...

Q5. The most common use of the one-dimensional array in C is the

a. String

b. Character

c. Data

Q6. C provides loop constructs for performing loop operations. They are

a. The WHILE statement

b. The DO statement

c. The FOR statement

### MCQs Answers

Q1. (c)

Q2. (c)

Q3. (b)

Q4. (a)

Q5. (c)

Q6. (b)

### References:

1. Sebesta, "Concept of programming Language", Pearson Edu
2. Loudon, "Programming Languages: Principles & Practices", Cengage Learning
3. Tucker, "Programming Languages: Principles and paradigms ", Tata McGraw -Hill.
4. E Horowitz, "Programming Languages", 2nd Edition, Addison Wesley

### Related posts:

1. PPL:Named Constants

2. Parse Tree | PPL | Prof. Jayesh Umre
3. Basic elements of Prolog
4. Loops | PPL | Prof. Jayesh Umre
5. Subprograms Parameter passing methods | PPL | Prof. Jayesh Umre
6. Programming Paradigms | PPL | Prof. Jayesh Umre
7. Subprograms Introduction | PPL | Prof. Jayesh Umre
8. Phases of Compiler | PPL | Prof. Jayesh Umre
9. Parse Tree | PPL
10. Influences on Language design | PPL | Prof. Jayesh Umre
11. Fundamentals of Subprograms | PPL | Prof. Jayesh Umre
12. Programming Paradigm
13. Influences on Language Design
14. Language Evaluation Criteria
15. OOP in C++ | PPL
16. OOP in C# | PPL
17. OOP in Java | PPL
18. PPL: Abstraction & Encapsulation
19. PPL: Semaphores
20. PPL: Introduction to 4GL
21. PPL: Variable Initialization
22. PPL: Conditional Statements
23. PPL: Array
24. PPL: Strong Typing
25. PPL: Coroutines
26. PPL: Exception Handler in C++
27. PPL: OOP in PHP
28. PPL: Character Data Type

29. PPL: Exceptions
30. PPL: Heap based storage management
31. PPL: Primitive Data Type
32. PPL: Data types
33. Programming Environments | PPL
34. Virtual Machine | PPL
35. PPL: Local referencing environments
36. Generic Subprograms
37. Local referencing environments | PPL | Prof. Jayesh Umre
38. Generic Subprograms | PPL | Prof. Jayesh Umre
39. PPL: Java Threads
40. PPL: Loops
41. PPL: Exception Handling
42. PPL: C# Threads
43. Pointer & Reference Type | PPL
44. Scope and lifetime of variable
45. Design issues for functions
46. Parameter passing methods
47. Fundamentals of sub-programs
48. Subprograms
49. Design issues of subprogram
50. Garbage Collection
51. Issues in Language Translation
52. PPL Previous years solved papers
53. Type Checking | PPL | Prof. Jayesh Umre
54. PPL RGPV May 2018 solved paper discussion| Prof. Jayesh Umre
55. PPL Viva Voce

- 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++