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 understand1

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-dimentional 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)

Principles of Programming Languages:

EasyExamNotes.com covered following topics in these notes.

- Language Evaluation Criteria
- Influences on Language Design
- Language Categories
- Programming Paradigms
- Compilation

- Virtual Machines
- Programming Environments
- Issues in Language Translation
- Parse Tree
- Pointer and Reference type
- Concept of Binding
- Type Checking
- Strong typing
- Sequence control with Expression
- Exception Handling
- Subprograms
- Fundamentals of sub-programs
- Scope and lifetime of variable
- static and dynamic scope
- Design issues of subprogram and operations
- Local referencing environments
- Parameter passing methods
- Overloaded sub-programs
- Generic sub-programs
- Design issues for functions
- co routines
- Abstract Data types
- Abstraction and encapsulation
- Static and Stack-Based Storage management
- Garbage Collection
- OOP in C++
- OOP in Java
- OOP in C#

- OOP in PHP
- Concurrency
- Semaphores
- Monitors
- Message passing
- Java threads
- C# threads
- Exception handling
- Exceptions
- Exception Propagation
- Exception handler in C++
- Exception handler in Java
- Introduction and overview of Logic programming
- Basic elements of Prolog
- Application of Logic programming
- Functional programming languages
- Introduction to 4GL

Practicals:

- Memory Implementation of 2D Array.
- Memory Implementation of 3D Array.
- Implementation of pointers in C++.
- Write a program in Java to implement exception handling.
- Write a program in C++ to implement call by value parameter passing Method.
- Write a program in C++ to implement call by reference parameter passing Method.
- Write a program in Java to implement concurrent execution of a job

using threads.
- Implement Inheritance in C#.
- Implement Encapsulation in C#.
- Implement static/compiletime Polymorphism in C#.
- Implement dynamic/runtime Polymorphism in C#.

Previous years solved papers:

- PPL|RGPV|May 2018
- PPL|RGPV|June 2017

A list of Video lectures

- Click here

References:

1. Sebesta,"Concept of programming Language", Pearson Edu
2. Louden, "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. 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 Viva Voce
57. PPL RGPV June 2017 Solved paper | Prof. Jayesh Umre
58. Concurrency
59. Basic elements of Prolog
60. Introduction and overview of Logic programming
61. Application of Logic programming
62. PPL: Influences on Language Design
63. Language Evaluation Criteria PPL
64. PPL: Programming Environments
65. PPL: Virtual Machine
66. PPL: Programming Paradigm
67. PPL: Pointer & Reference Type
68. try-catch block in C++