|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 Introduction to C |  | 7.2.3 Structure variable initialization : |  | 13.7 Handling exceptions thrown by new |  |
| 1.1 The components of a C program |  | 7.2.4 Accessing members of a structure & use of "." operator |  | 13.8 RTTI (run-time type identification) |  |
| 1.2 DECLARE VARIABLES AND ASSIGN VALUES |  | 7.2.5 Structures as arrays : |  | 13.9 C++ casting operators |  |
| 1.3 Properties of function and C keywords |  | 7.2.6 Arrays within Structures : |  | 13.9.1 dynamic\_cast: |  |
| 1.4 USE FUNCTIONS TO RETURN VALUES |  | 7.2.7 COPYING AND COMPARING STRUCTURE VARIABLES |  | 13.9.2 const\_cast, reinterpret\_cast and static\_cast |  |
| 1.5 Arguments of functions and their use |  | 7.2.8 Structures and Functions |  |  |  |
| 1.6 REMEMBER THE C KEYWORDS |  | 7.2.9 SIZE OF STRUCTURES |  | Miscellaneous topics & STL |  |
| 2 C Control Statements |  | 7.5 ENUMERATIONS |  | 14.1 namespace Details |  |
| 2.1 The Block of statements or Code Bolcks |  | 7.6 typedef |  | 14.2 Conversion function (CvF) |  |
| 2.2 The Selection statements |  | 7.7 Bitwise and Shift Operators |  | 14.3 static Class Members |  |
| 2.3 The "if" statement |  | 7.8 OPERATORs Advanced |  | 14.4 const MEMBER FUNCTIONS AND mutable |  |
| 2.4 The "if-else" statement |  | 7.8.2 The Comma Operator : |  | 14.5 Initializing object using "=" and the "explicit" specifier |  |
| 2.5 The "for" loop |  | 7.8.3 More Uses Of Assignment Operator |  | 14.6 LINKAGE specifier for linking other language. asm for linking assembly language. |  |
| 2.6 The increment operator "++" and decrement operator " - - " |  | 7.8.4 The precedence of all C - OPERATORS |  | 14.7 ARRAY-BASED I/O (Not will be used) |  |
| 2.7 The backslash charecters \n "newline", \t "tab" etc |  |  |  | Standard Template Library (STL) |  |
|  |  | 8 C Preprocessors and Advanced topics |  | 14.8 An Overview Of STL |  |
| 2.8 RELATIONAL AND LOGICAL OPERATORS |  | 8.1 Advanced #define and #include |  | 14.9 Type Names (Placeholder Types) For Container Classes |  |
| NOTE : Execution of printf with ++ operators |  | 8.2 Conditional COMPILATlON |  | 14.10 VECTORS |  |
| 2.9 Character inputs from keyboard |  | 8.3 #error, #undef, #Iine, #pragma |  | 14.11 LISTS |  |
| 2.10 NEST if statements and if-else-if LADDER |  | 8.4 C's built-in MACROS |  | 14.12 MAPS (example of an associative container) |  |
| 2.11 for Loop – 'Advanced' |  | 8.5 The # and ## operators |  | 14.13 ALGORITHMS (names of the algorithms with purpose) |  |
| 2.12 while Loop |  | 8.6 DYNAMIC ALLOCATION |  | 14.14 STRING class |  |
| 2.13 do Loop |  |  |  |  |  |
| 2.14 NESTED LOOPS |  | 9 Introduction to C++ |  |  |  |
| 2.15 Use break TO EXIT A LOOP |  | 9.1 Get familiar with OOP (Object Oriented Programming) |  |  |  |
| 2.16 Use continue for skipping any code |  | 9.2 Old Header And Standard Header Declaration Of C++ |  |  |  |
| 2.17 The switch STATEMENT |  | 9.3 Function overloading in C++ |  |  |  |
| 2.18 The goto STATEMENT |  | 9.4 C++ comments |  |  |  |
| 2.19 C control statements |  | 9.5 C++ Console I/O |  |  |  |
|  |  | 9.6 Difference between C and C++ |  |  |  |
| 3 Advanced data types, variables & Expression |  | 9.7 C++ Keywords |  |  |  |
| 3.1 DATA - TYPE MODIFIERS |  |  |  |  |  |
| 3.2 Advanced variable declaration (Local and Global) |  | 10 C++ Class, Objects with array and pointer |  |  |  |
| 3.3 Constants Advanced |  | 10.1 Introduction to CLASS |  |  |  |
| 3.4 Variable initialization |  | 10.2 CONSTRUCTOR and DESTRUCTOR Functions |  |  |  |
| 3.5 Type conversion in "Expression" and "Assignment" |  | 10.3 Constructors with Parameters |  |  |  |
| 3.6 The TYPE CASTS |  | 10.4 Relation between STRUCTURES-UNIONS and CLASSES |  |  |  |
|  |  | 10.5 In-Line Functions & Automatic In-Lining |  |  |  |
| 4 Strings, Arrays & Pointers |  | 10.6 Assigning Objects |  |  |  |
| 4.1 One dimensional Arrays |  | 10.7 Object Pointers |  |  |  |
| 4.2 USE STRINGS : gets(), 4-string functions, atoi(), STRING.H & STDLIB.H |  | 10.8 The "this" pointer |  |  |  |
| 4.3 Create multidimensional Arrays |  | 10.9 ARRAYS OF OBJECTS |  |  |  |
| 4.4 Initialize Arrays |  | 10.10 PASSING objects to functions and RETURNING objects from function |  |  |  |
| 4.5 ARRAYS OF STRINGS |  | 10.11 Memory allocation/release operators : new, delete |  |  |  |
| 4.6 The POINTERS |  | 10.12 References |  |  |  |
| 4.7 Restriction to Pointer Expression |  |  |  |  |  |
| 4.8 POINTERS WITH ARRAYS |  | Overloading: function & Operators. |  |  |  |
| 4.9 Use pointers to string constants |  | Inheritance. |  |  |  |
| 4.10 Arrays of Pointers |  | 11.1 Introduction to INHERITANCE |  |  |  |
| 4.11 Multiple INDIRECTION |  | 11.2 Intro to FRIEND functions |  |  |  |
| 4.12 Pointers as Parameters |  | 11.3 Overloading CONSTRUCTOR |  |  |  |
|  |  | 11.4 COPY CONSTRUCTOR (recall 10.10) |  |  |  |
| 5 Functions in C |  | 11.5 Default arguments |  |  |  |
| 5.1 The PROTOTYPE |  | 11.6 Ambiguity Caused By Overloading |  |  |  |
| 5.2 Recursion |  | 11.7 Address of an OVERLOADED function (recall 5.8) |  |  |  |
| 5.3 Parameters Advanced |  | 11.8 Overloading MEMBER OPERATOR FUNCTIONS |  |  |  |
| 5.4 Pass Arguments To Main() |  | 11.9 Overloading Binary Operators |  |  |  |
| 5.5 Old-Style Parameter Declarations |  | 11.10 Overloading the RELATIONAL and LOGICAL operators |  |  |  |
| 5.6 variable storage class specifire (Advanced topic) |  | 11.11 Overloading A UNARY Operator |  |  |  |
| 5.7 Access modifiers : const and volatile (Advanced topic) |  | 11.12 Overloading FRIEND OPERATOR FUNCTIONS |  |  |  |
| 5.8 Function Pointers (Advanced topic) |  | 11.13 Assignment Operator Advanced |  |  |  |
|  |  | 11.14 Overloading The [ ] SUBSCRIPT Operator |  |  |  |
| 6 C's Console input/output & File input/output |  | 11.15 INHERITANCE: access control of base class |  |  |  |
| 6.5 Details on gets() and puts( ) |  | 11.16 Accessing PROTECTED members |  |  |  |
| 6.6 printf( ) : Details |  | 11.17 INHERITANCE with Constructors-Destructors |  |  |  |
| 6.7 scanf( ) : Details |  | 11.18 MULTIPLE INHERITANCE |  |  |  |
| 6.8 STREAMS for file I/O in C |  | 11.19 VIRTUAL BASE (problems with "one derived" & "multiple direct base") |  |  |  |
| 6.9 File access using fopen(), fclose() and read/write using fgetc(), fputc() |  |  |  |  |  |
| 6.10 End of file [EOF] feof() And file error checking ferror() |  | C++ I/O system |  |  |  |
| 6.11 String I/O in a File with fputs() & fgets(). Text I/O with fprintf() & fscanf() |  | 12.1 C++ I/O Stream |  |  |  |
| 6.12 READ AND WRITE BINARY DATA |  | 12.2 Formatted I/O |  |  |  |
| 6.13 Random access using fseek() |  | 12.3 width(), precision(), AND fill() |  |  |  |
| 6.14 Some other important File-System functions |  | 12.4 I/O MANIPULATORS |  |  |  |
| 6.15 THE STANDARD STREAMS |  | 12.5 Inserters and Extractors |  |  |  |
|  |  | 12.6 User Defined Manipulators |  |  |  |
| 7 Custom (User-defined) data-types & |  | 12.7 File I/O |  |  |  |
| Advanced Operators |  | 12.8 UNFORMATTED I/O & BINARY I/O |  |  |  |
| 7.1 The Custom (User-defined) Data-types of C |  | 12.9 Checking I/O Status |  |  |  |
| 7.2 STRUCTURE Basics |  | 12.10 Random Access |  |  |  |
| 7.2.1 Defining structures |  | 12.11 Customized I/O And Files |  |  |  |
| 7.2.2 Declaring structure variables : |  |  |  |  |  |
|  |  | Polymorphism, Exceptions, RTTI, Operator cast |  |  |  |
|  |  | 13.1 Pointers To Derived Classes |  |  |  |
|  |  | 13.2 Virtual Functions (VF) |  |  |  |
|  |  | 13.3 Abstract class and Pure Virtual function (PVF) |  |  |  |
|  |  | 13.4 Polymorphism: Early binding & Late binding |  |  |  |
|  |  | 13. 5 Generic-Functions & Generic-Classes (GnF & GnC) |  |  |  |
|  |  | 13.6 EXCEPTION HANDLING |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |