KENDRIYA VIDYALAYA SANGATHAN, MUMBAI REGION

SPLIT UP SYLLABUS (2017 – 18)

SUBJECT : COMPUTER SCIENCE WITH C++

CLASS : XI

UNIT NO.	UNIT NAME	MARKS	MINIMUM THEORY PERIODS	MINIMUM PRACTICAL PERIODS
1.	COMPUTER FUNDAMENTALS	10	18	06
2.	PROGRAMMING METHODOLOGY	12	28	10
3.	INTRODUCTION TO C++	14	44	36
4.	PROGRAMMING IN C++	34	50	48
	TOTAL	**70**	**140**	**100**

MONTHWISE SPLITUP (2017 – 18)

MONTH	CONTENTS	APPRX. NO. OF PERIODS REQUIRED (THEORY)	APPRX. NO. OF PERIODS REQUIRED (PRACTICAL)	PRACTICALS.
JUNE 2017	**UNIT 1: COMPUTER FUNDAMENTALS** Classification of Computers: Basics of computer and its operation; Functional Components; Concept of Booting. Software concepts: Types of Software	06	04	

JULY 2017	**Software Concepts:**
	Types of Software - System Software, Utility Software and Application Software;
	System Software: Operating System, Compilers, Interpreters and Assembler;
	Operating System: Need for operating system, Functions of Operating System (Processor Management, Memory Management, File Management and Device Management), Types of Operating system – Interactive (GUI based), Time Sharing, Real Time and Distributed; Commonly used operating systems: Solaris, UNIX,LINUX, Mac OS, MS Windows; Mobile OS- Android, Symbian.
	Utility Software: Antivirus, File Management Tools, etc.
	Open Source Concepts: Open Source Software, Freeware, Shareware, Proprietary Software
		23	12	Familiarization with C++ Editor;
				Writing basic
				Program of the C++ program;
				Small Programs to understand C++ Editor and Compiler

	Application Software: Office Tools, Domain Specific Tools **Number System:** Binary, Octal, Decimal, Hexadecimal and conversion amongst these number system. **Internal Storage encoding of characters:** ASCII, ISCII (Indian Script standard code for Information Interchange), and UNICODE(for Multilingual Computing). **Microprocessor:** **Basic** concepts, clock speed(MHz, Ghz), 16 bit, 32 bit, 64 bit, 128 bit processors. Types – RISC, CISC, EPIC **Memory Concepts:** Units, Primary Memory – Cache, RAM, ROM; Secondary **Memory –** Fixed and Removable Storage, I/O Ports and Connections			

AUGUST 2017	**1st Periodic Test**
	UNIT 2: PROGRAMMING METHODOLOGY
	General Concepts: Modular approach; Clarity and Simplicity of Expressions, Use of proper Names for identifiers, Comments, Indentation; Documentation and Program Maintenance; Running and Debugging programs, Syntax Errors, Run-Time Errors, Logical Errors;
	Problem Solving Methodology: Understanding of the problem, Identifying minimum number of inputs required for output, Step by step solution for the problem, breaking down solution into simple steps, Identification of arithmetic and logical operations required for solution, Using Control Structure: Conditional control and looping (finite and infinite);
	Problem Solving: Introduction to Algorithm/FlowCharts
	
	UNIT 3: Introduction to C++
	Getting Started with C++:
	C++ character set, C++ Tokens (Identifiers, Keywords, Constants, Operators), Structure of a C++ Program (include files, main function); Header files – iostream.h, iomanip.h; **cout**, **cin**; Use of I/O operators (<< and >>), Use of endl and setw(), Cascading of I/O operators, Error Messages; Use of editor, basic commands of editor, compilation, linking and execution; standard input/output operations from C language: gets(), puts() of stdio.h header file;
	
	Data Types, Variables and Constants:
	Concept of Data types; Built-in Data types: **char**, **int**, **float** and **double**; Constants: Integer Constants, Character Constants (Backslash character constants - \n, \t), Floating Point Constants, String Constants; Access modifier: **const**; Variables of built-in data types, Declaration/Initialisation of variables, Assignment statement; Type modifier: signed, unsigned, long;
		21	10	Programs to use Cascaded I/O Operators;
				
				Programs to understand Data Types;
				
				Programs to understand concept of Variable and Constants
				
				Programs to understand Operators and Expressions

SEPTEMBER 2017	**UNIT 3: Introduction to C++ (Continued)** **Operators and Expressions:** Operators: Arithmetic operators (-,+,*,/,%), Unary operator (-), Increment and Decrement Operators (—,++), Relational operators (>,>=,<,<=,==,!=), Logical operators (!, &&, ||), Conditional operator: <condition>?<if true>:<else>; Precedence of Operators; Expressions; Automatic type conversion in expressions, Type casting; C++ shorthand's (+=, -=, *=, /=, %=); **UNIT 4 : PROGRAMMING IN C++** **Flow of control:** **Conditional statements: if-else**, Nested **if**, **switch..case..default**, Nested **switch..case**, break statement (to be used in switch..case only); **Loops: while**, **do - while** , **for** and Nested loops;	20	16	Programs based on – Selection Construct & Iteration Construct
OCTOBER 2017	**Half Yearly Exam** **UNIT 4: PROGRAMMING IN C++ (Continued)** **Header Files and Inbuilt Functions:** stdio.h, ctype.h, string.h, math.h **String Functions:** **Header File:** string.h **Functions:** isalnum(), isalpha(), isdigit(), islower(), isupper(), tolower(), toupper(); **Character Functions:** **Header File**: ctype.h **Functions:** isalnum(), isalpha(), isdigit(), islower(), isupper(), tolower(), toupper(), strcpy(), strcat(), strlen(), strcmp(), strcmpi(); **Mathematical Functions:** **Header File**: math.h, stdlib.h; **Functions**: fabs(), log(), log10(), pow(), sqrt(), sin(), cos(), abs(), **Other Functions:** **Header File**: stdlib.h; **Functions:** randomize(), random(); **Introduction to User Defined Functions and its requirements** Defining a function; function prototype, Invoking/calling a function, passing arguments to function, specifying argument data types, default argument, constant argument, call by value, call by reference, returning values from a function, calling functions with arrays,	16	10	Programs to understand Functions - Passing values/parameters - Returning values/parameters - Call By Value/Call By Reference
NOVEMBER 2017	**Introduction to User Defined Functions (CONTD.)** Scope rules of functions and variables; local and global variables; Relating to Parameters and return type concepts in built-in functions	24	16	Programs to understand Scope Rules

	Structured Data Type: Array Declaration/initialization of **One-dimensional array**, Inputting array elements, Accessing array elements, Manipulation of Array elements (sum of elements, product of elements, average of elements, linear search, finding maximum/minimum value); Declaration/Initialization of a String, string manipulations (counting vowels/consonants/digits/ special characters, case conversion, reversing a string, reversing each word of a string);			Programs to understand 1-d Array, Strings
DECEMBER 2017	**Two-dimensional Array:** Declaration/initialisation of a two-dimensional array, inputting array elements Accessing array elements, Manipulation of Array elements (sum of row element, column elements, diagonal elements, finding maximum/minimum values);	18	8	Program On array of string; Programs Based on 2-d array Programs using Function and Array together
JANUARY 2018	**2nd Periodic Test** **STRUCTURES** **Introduction** Referencing structure Elements Nested structure Structure and Array USER DEFINED Data type #Define Preprocessor directives.	24	12	Programs to understand Concept of Fields, Passing Objects in Function, Returning Objects from Functions
FEBRUARY 2018	**REVISION**	23	12	Project Work
MARCH 2018	**SESSION ENDING EXAM & RESULT**	25	---	**----**