

YEAR PLAN 2018 – 2019
Grade XII COMPUTER SCIENCE

The academic year is divided into three sessions.

Session One: June to August 2018.

Session Two: September 2018 to December 2018.

Session Three: January 2019

Cumulative Assessment(End of month): June, July, October

Summative Assessment I – August 2018

Model Exam-I-December 2018

Model Exam -II – January 2019

AIM: To understand algorithmic problem solving using data abstractions, functional and procedural abstractions, and object based and object oriented abstractions.

To understand: (a) how computers represent, store and process data by studying the architecture and machine language of a simple microprocessor and different levels of abstraction that mediate between the machine and the algorithmic problem solving level and (b) how they communicate with the outside world.

ENDURING UNDERSTANDING: Any given problem may have different solutions.

GENERAL OBJECTIVE: To gain the ability to logically analyse problems and derive the most optimum solution.

Duration	Topics/Units	Specific Learning Objectives	Activities	Resources
Summer Class 19/3 to 6/4	Inheritance and Polymorphism	Understand the concept and do programming on Inheritance; base and derived classes; member access in derived classes; redefinition of variables and functions in subclasses; abstract classes; class Object; protected visibility. Interfaces, (definition and implementation), Subclass polymorphism and dynamic binding.	<ul style="list-style-type: none"> Notes Problem solving Text book exercises 	<ul style="list-style-type: none"> Teacher's notes Question Bank on every topic Past examination papers. Reference Textbooks by Dheeraj Malhotra, by Pandey and Dey.
June	Basic input/output operations on files and streams Simple Data Structures (Stacks, and Queues) Monthly Test	Reduce Boolean expressions to POS and SOP forms; Karnaugh's maps (up to four variables). Learn to implement data structures like Stacks, Queues, Circular Queues and Dequeues Learn the difference between array and string, Manipulate character sets using the accessor methods of the following classes: a) String b) StringBuffer c) StringTokenizer Perform basic input/output using Scanner and Printer classes from JDK; extract tokens from an input stream using StringTokenizer and StringTokenizer.	<ul style="list-style-type: none"> Notes on simplification procedure and rules. Problems on K-maps Accessor methods of String class Exercises from the text books Programming Worksheets and tests 	<ul style="list-style-type: none"> Guides – Oswal Question Series. Vatsal Question Series. Websites : icseguess.com , guideforschool.com, examfear.com
July	Propositional logic, boolean algebra and hardware Introduction to Data Structures Monthly Test	Learn to use basic data structures (linked lists) and implement them directly through classes	<ul style="list-style-type: none"> Worksheets and Tests Algorithms on basic data structures Programming Exercises from the text books 	<ul style="list-style-type: none"> Teacher's notes

August	Revision First Terminal Examination		<ul style="list-style-type: none"> Worksheets and Tests Notes on key terms. Programming Exercises from the text books 	<ul style="list-style-type: none"> Question Bank on every topic Past examination papers. Reference Textbooks by Dheeraj Malhotra, by Pandey and Dey. Guides – Oswal Question Series. Vatsal Question Series. Websites : icseguess.com , javaforschool.com, examfear.com
September	Recursion	Concept of recursion, simple recursive functions (e.g. factorial, GCD, binary search, conversion of representations of numbers between different bases). Recursive sorting techniques.	<ul style="list-style-type: none"> Worksheets and Tests Stack Tracing Writing code snippets Programming Exercises from the text books 	
October	Propositional logic, boolean algebra and hardware Recursive Data Structures Monthly Test	Data structures defined as abstract data types with a well defined interface Simplest and most efficient algorithms for recursive data structures The following should be covered for each data structure: Linked List (single): insertion, deletion, reversal, extracting an element or a sub list, checking emptiness. Binary trees: definition, external and internal nodes, height, completeness, balancing, traversals (pre, post and in-order).	<ul style="list-style-type: none"> Worksheets and Tests Algorithms on Lists and trees. Writing code snippets Programming Exercises from the text books 	
November	Propositional logic, Boolean algebra and hardware Computational Complexity Revision	Concrete computational complexity; concept of input size; estimating complexity in terms of functions; importance of dominant term; best, average and worst case. Big O notation for computational complexity; analysis of complexity of example algorithms using the big O notation (e.g. Various searching and sorting algorithms, algorithm for solution of linear equations etc.).	<ul style="list-style-type: none"> Worksheets and Tests Notes on key terms Problem solving Exercises from the text books 	
December	Model Exam I Xmas Holidays	Remedial, Revise, Tests		
January	Model Exam II	Remedial		
February/March	Board Exams			

Projects/ Field trips: As per the guidelines given by ISC

Text Book : Grade XII Computer Science by Sumita Arora (Dhanpat Rai & Co.)

Facilitators' name: Mrs Armin Shroff, Mrs. Kavitha M S

Blog: computerapplication976.wordpress.com for updates on worksheets and sample papers.

Approved by the Principal