COMPUTER SCIENCE (71)

Aims:

1. To enable candidates to comprehend the concepts and practices of Computer science.
2. To develop an understanding of how computers store and process data.
3. To enable candidates to describe the major components of computer hardware, their functions and interaction.
4. To develop an understanding of the fundamental concepts of programming and the ability to apply the same.
5. To develop an appreciation of the implications of computer use in contemporary society.

CLASS IX

*There will be **one** paper of **two** hours duration carrying 80 Marks and Internal Assessment of 20 Marks.*

*The paper will be divided into **two** Sections A and B.*

***Section A** (20 marks): This section will consist of compulsory short answer questions, testing knowledge, application and skills relating to elementary/fundamental aspects of the entire syllabus.*

***Section B** (60 marks): This section will consist of questions based on programming. There will be a choice of questions and candidates will be required to answer **four** questions from this section.*

PART I – THEORY

1. Computer hardware: parts of a computer and their functions

CPU, the clock, cache memory, primary memory, secondary memory, input and output devices, communication devices (the aim is not to describe/discuss an exhaustive list of devices but to understand what parts are present in a typical computer and what the function of each part is).

2. Data representation and internal computer structure

(i) Number systems, base of a number system - decimal, binary, octal, hexadecimal representation, conversion between various representations, character representations (ASCII, ISCII, Unicode).

(ii) Representations for integers, real numbers, limitations of finite representations.

(iii) Internal structure of a computer, a simple decimal load and store computer and its machine language, instruction format, registers, program counter, instruction register; register addressing modes, instruction cycle, assembly language for the same computer, simple algorithms in assembly language.

3. Computer software

The boot process, operating system (resource management and command processor), file system.

(i) Boot process, operating systems - resource management, command processing.

(ii) Directories, files and hierarchical file system.

(iii) Programming languages (machine language, assembly language, high level language).

(iv) Compilers and interpreters.

(v) Application software.

4. Social context of computing and ethical issues

(i) Intellectual property and corresponding laws and rights, software as intellectual property.

(ii) Software patents, copyrights, and trademarks, software licensing and piracy.

(iii) Free software foundation and its position on software, open source software.

(iv) Privacy, email etiquette.

5. Algorithms

(i) Concept of an algorithm.

(ii) Properties of an algorithm (finite, definite, terminating, precise).

(iii) Basic ideas of the complexity of an algorithm - space complexity, time complexity.

6. Programming using a High Level Language

The programming element in the syllabus is aimed at problem solving and **not** on merely rote learning of the commands and syntax of particular programming languages, Students have the option to use either BASIC or C++ in order to implement the high level language concepts and algorithms and to use them for solving problems. While choosing BASIC care must be taken to choose a standard version that has "block if structures", "functions through which parameters may be passed and values returned". **Very old versions using "goto statements" must not be used.** Care must be taken that **'standard and recent'** versions of the languages are used on the computer. It is recommended that students mention the version of the language being used while writing answers in order to avoid ambiguity. For example, software such as Microsoft Quick BASIC, Borland Turbo **C++**, **Visual C++** or **GNU C++** on **Linux** can be used.

(i) Primitive data types supported by the language (integers, floating point numbers, characters, booleans etc. - will depend on the language), variables (and their declaration – based on language), assignment, difference between the left-hand side and right-hand side of an assignment.

(ii) Expressions - arithmetic and logical, evaluation of expressions, type of an expression (depends on language). Operators, associativity and precedence of operators.

(iii) Statements, blocks (where relevant), scope and visibility of variables.

(iv) Conditional statements (if and if-then-else), switch, break, default.

(v) Loops (for, while-do, do-while).

(vi) Simple input/output using standard input/output.

7. Computers in everyday life

(i) Familiarity with software for word processing, databases, spreadsheets, making presentations.

(ii) Basic introduction to the Internet, browsing, email.

PART II - INTERNAL ASSESSMENT (PRACTICAL WORK)

Part II (Practical work) will carry 20 marks and shall be assessed on a continuous basis throughout the year. The assessment of practical work should include small projects using software in item 7 and solutions to programming problems in item 6, which have been coded and run in the higher level language being used in the course.

Teachers should maintain a record of work done through the year and give it due credit at the time of cumulative evaluation at the end of the year.

CLASS X

*There will be **one** paper of **two** hours duration carrying 80 Marks and Internal Assessment of 20 Marks.*

*The paper will be divided into **two** Sections A and B.*

***Section A** (20 marks): This section will consist of compulsory short answer questions, testing knowledge, application and skills relating to elementary/fundamental aspects of the entire syllabus.*

***Section B** (60 marks): This section will consist of questions based on programming. There will be a choice of questions and candidates will be required to answer **four** questions from this section.*

PART I – THEORY

1. Computer Structure

(i) Logic gates (NOT, AND, OR, XOR) and their use in computers.

(ii) Review of number systems (binary, decimal, octal, hexadecimal), representation for different types - integers, float, characters.

(iii) Simple binary arithmetic, including addition, subtraction, multiplication and division.

(iv) Computer logic, Boolean operations, logical operators (NOT, AND, OR, XOR) and their truth tables.

2. Review of Programming

Review of programming in BASIC or in C++ from Class IX.

(i) Primitive data types supported by the language (integers, floating point numbers, characters, booleans etc. - will depend on the language), variables (and their declaration - based on language), assignment, difference between the left-hand side and right-hand side of an assignment.

(ii) Expressions - arithmetic and logical, evaluation of expressions, type of an expression (depends on language). Operators, associativity and precedence of operators.

(iii) Statements, blocks (where relevant), scope and visibility of variables.

(iv) Conditional statements (if and if-then-else), switch, break, default.

(v) Loops (for, while-do, do-while).

(vi) Simple input/output using standard input/output.

3. Advanced Programming

The programming element in the syllabus is aimed at problem solving and not on merely rote learning of the commands and syntax of particular programming languages. Students have the option to use either BASIC or C++ in order to implement algorithms and to use them for solving problems. While choosing BASIC, care must be taken to choose a standard version that has "block if structures", "functions through which parameters may be passed and values returned". **Very old versions using "goto statements" must not be used.** Care must be taken that **'standard and recent'** versions of the languages are used on the computer - it is recommended that students mention the version of the language being used while writing answers in order to avoid ambiguity. For example, software such as Microsoft Quick BASIC, Borland Turbo C++, Visual C++ or GNU C++ on Linux can be used.

(i) Functions / subroutines as procedural abstractions. Using functions / subroutines in programs.

(ii) Arguments and argument passing in functions/subroutines.

(iii) Scope of variables.

(iv) Structured types, arrays as an example of a structured type. Use of arrays in sorting and searching. Two-dimensional arrays. Use of two-dimensional arrays to represent matrices. Matrix arithmetic using arrays. Use of arrays to solve linear equations (Gauss elimination method).

(v) Review of input/output using standard input and standard output from Class IX. Input/output using sequential files. Opening, closing files. Creating and deleting files. Formatting output. Concept of a token and separator. Extracting tokens from the input.

(vi) Characters, ASCII representation, strings as a composite data type; functions on strings (ex. length, substring, concatenate, equality, accessing individual characters in a string, inserting a string in another string at a given location).

(vii) Simple type casting for primitive types; inter-conversion between character/string types and numeric types.

(viii) Distinction between compile time and run time errors. Run time errors due to finite representations - overflow, underflow. Other run time errors.

(ix) Basic ideas about linking, loading, execution.

4. Documentation of programs

Need for good documentation; good documentation practices; standards and naming conventions.

5. Practical Work

Regular programming in labs should supplement every topic that is taught in the classroom. The students will be expected to invent algorithmic solutions expressed in C++ or Basic to solve problems and then actually implement and run the program to get answers.

The student will also be required to do a project that involves significant programming effort.

PART II - INTERNAL ASSESSMENT

(PRACTICAL WORK)

Internal Assessment will comprise of assignments and Project work.

Minimum number of Assignments

Assignments as prescribed by the teacher to cover all the concepts in the programming syllabus.

Project Work - One project in BASIC or C++.

Suggested Assignments

- The generation of all three-digit prime numbers, mensuration, calculating income tax, commissions, etc.
- Various number problems, twin primes, perfect numbers, Syracuse numbers, etc.
- A programming assignment on a problem resulting in formatted screen or printer output, Example: forming a diamond shape with the* character in the middle of the screen, calculating and outputting electricity bills, printing tabular data.
- A small menu driven program, marks processing and ranking algorithms.
- Sorting and searching algorithms.
- Permutation generation algorithms.
- Finding mode, mean, median of a set of numbers.
- File handling assignment. A sequential file may be prepared by the teacher to be manipulated by the students in one or more of the following ways: Reading and filtering data according to given criteria, adding/deleting/ modifying records.

Programming Project

Proposed Guidelines for Marking

The teacher should use the criteria below to judge the internal work done. Basically, four criteria are being suggested: Analysis, Algorithm Design, Coding and Documentation and Execution. The important questions to be asked when evaluating each criterion are shown. 25% of the total credit is assigned to each criterion - so each is equally important. The actual grading will be done by the teacher based on his/her judgment. However, one recommended criteria is: divide the outcome for each criterion into one of 4 groups: excellent, good, fair/acceptable, poor/ unacceptable, then use numeric values for each grade and add to get the total which can be multiplied by a suitable factor to get the final marks.

Analysis:

Has the problem been analyzed carefully?

Are all attributes with the right kinds of types present?

Has the problem been broken up into proper segments?

Algorithm design:

Is the choice of data structures proper?

Is the algorithm suitable for the problem?

How efficient is it?

Coding and Documentation:

Is the coding done properly? (Choice of names, no unconditional jumps, proper organization of conditions, proper choice of loops, error handling, code layout)

Is the documentation complete and readable? (Documentation, variable documentation, function documentation, constraints, known bugs - if any)

Execution:

Does the program run correctly on all sample input?

Criteria (total marks - 20)	Analysis (mm-5)	Algorithm Design (mm-5)	Coding and Documentation (mm-5)	Execution (mm-5)
Excellent	5	5	5	5
Good	4	4	4	4
Fair	3	3	3	3
Poor	2	2	2	2

EVALUATION

Teachers should maintain a record of work done through the year and give it due credit at the time of cumulative evaluation at the end of the year.

An External Examiner shall evaluate the one project built by the candidates. The examiner shall view the project and conduct a viva to judge the depth of knowledge and understanding of the candidate.

An External Examiner shall be nominated by the Principal and may be a teacher from the faculty, but not teaching the subject in the relevant section/class. For example, a teacher of Computer Science of Class VIII may be deputed to be the External Examiner for Class X, Computer Science Projects.

Evaluation of practical work will be done as follows:

Award of Marks (20 Marks)

Subject Teacher (Internal Examiner) 10 Marks

External Examiner 10 Marks

The total marks obtained out of 20 are to be sent to the Council by the Principal of the school.

The Head of the school will be responsible for the entry of marks on the mark sheets provided by the Council.

EQUIPMENT

There should be enough computer systems to provide for a teaching schedule where at least three-fourths of the time available is used for programming and project work.

The hardware and software platforms should be such that the students can comfortably develop and run programs on those machines.

Since hardware and software evolve and change very rapidly the schools shall need to upgrade them as required.

Following are the **recommended** specifications as of now:

THE FACILITIES:

- A lecture cum demonstration room with a MULTIMEDIA PROJECTOR/ an LCD and O.H.P. attached to the computer.
- A white board with white board markers should be available.
- A fully equipped Computer Laboratory that allows one computer per student.
- Internet connection for accessing the World Wide Web and email facility.
- The computers should have a Minimum of 128 MB RAM and a PIII or Equivalent Processor.
- Good quality printers.
- A scanner, a web cam/a digital camera - should be provided if possible.

SOFTWARE:

There is a wide variety of software packages and operating systems and compiler available but software has to be chosen very carefully. Any suitable Operating System or Software Package, which is being used currently and is likely to be used in future, can be chosen.

The criteria used in the selection of software should be:

- It should have a good user interface so that the beginners may learn to use it easily.
- It should be used widely and be easily available.
- The material related to the software should be abundantly available.

In this respect the latest versions of the chosen software should be made available.

Great emphasis should be placed on ethics. Some people do not object to using pirated software. They do not realize that it has something to do with ethics. It is important to introduce these concepts to the students in the very beginning.

The Council does not recommend any Operating systems or Software Packages by any particular Vendor.

The schools are free to use Software Packages available in the Public Domain Software.