

## COMPUTER APPLICATIONS (86)

### Aims:

1. To empower students by enabling them to build their own applications.
2. To introduce students to some effective tools to enable them to enhance their knowledge, broaden horizons, foster creativity, improve the quality of work and increase efficiency.
3. To enable students to learn to use the World Wide Web in order to gather knowledge and communicate with students and the academic community all over the world.
4. To enable students to learn to process words and numbers, analyze data, communicate ideas effectively and make the optimum use of computer resources.
5. To help students learn fundamental concepts of computing using object oriented approach in one computer language.
6. To provide students with a clear idea of ethical issues involved in the field of computing.

### CLASS IX

*There will be **one** written paper of **two hours** duration carrying 100 marks and Internal Assessment of 100 marks.*

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

***Section A** (Compulsory – 40 marks) will consist of compulsory short answer questions covering the entire syllabus.*

***Section B** (60 marks) will consist of questions which will require detailed answers and there will be a choice of questions in this section*

### THEORY – 100 Marks

#### 1. Operating System

##### i) Introduction to Operating System

The need for an operating system, features and functions of an operating system and familiarity with installing and uninstalling software.

*Functions and features of an operating system (examples of single user and multiple users); familiarize the students with installation and un-installation of connected peripherals and other related softwares like dictionaries, encyclopedias, etc.*

##### ii) Graphic User Interface

Working with the Graphical User Interface (GUI), Elements of a GUI, handling files and directories under GUI, managing the desktop.

*Concept of an active window. Familiarity with the icons, the buttons and the task bar.*

*Resizing a window. Handling multiple windows.*

*Creating folders. Copying items. Moving items. Deleting Items. Setting attributes of items.*

*Creating shortcuts on the desktop, arranging the folders on the desktop.*

#### 2. The Internet

Internet, e-mail, WWW, modem and related protocols.

*Working of internet, creating an e-mail account, sending and receiving e-mails, web servers, search engines to perform a simple search and Boolean operators to fine tune a search, basic working of a modem (modulation and demodulation) through block diagrams only. Protocols and its need (FTP, HTTP, IMAP, POP, SMTP). Downloading information.*

#### 3. Computing and Ethics

Ethical issues in computing.

*Intellectual property rights; protection of individual's right to privacy; data protection on the internet; protection against Spam; software piracy, cyber crime, hacking, protection against malicious intent and malicious code.*

*The stress should be on good etiquette and ethical practices.*

#### 4. Office Application Software:

Word processor, Multimedia presentation and Spreadsheets.

*Discuss the following features for Word Processor, Multimedia Presentation and Spreadsheets:*

*Opening and closing, saving, editing, formatting, printing, spell check, grammar and header/footer.*

##### **Special features:**

**Word Processor** – *inserting a table and working with tables.*

**Multimedia Presentation** – *the students should be able to create a presentation using images / pictures, sound, video and custom animation. Students should be able to import material from word processors, spreadsheets, databases and internet.*

**Spreadsheets** – *Using formulae functions (mathematical), working with ranges, graphs and charts. Difference between absolute and relative references.*

#### 5. Database Packages

The need for database management; creating and saving a database; editing a database; performing calculations; modifying the structure of a database; sorting, indexing; querying; report generation. Working with multiple databases, object linking and embedding, creating applications.

*The need for database management for handling vast amount of data- storing, sorting, summarizing, classifying and retrieving quickly.*

*Defining the structure of a database, entering data of various types, saving it in an appropriate area.*

*Adding, deleting and modifying records, global editing.*

*Performing calculations on one record or a group of records.*

*Modifying the structure of a database by inserting, deleting or modifying fields.*

*Sorting on one field/ multiple fields, sorting selected records/ all the records.*

*Indexing on one field/ multiple fields. The need for re- indexing. Sorting vs. Indexing.*

*Setting query condition, Relational and Logical Operators, setting query using multiple conditions.*

*-Generating detailed or summary reports.*

*-Working with multiple databases and explain the relationship (one to one, one to many, many to one and many to many) through real life examples.*

*Linking objects/embedding objects, linking vs. embedding.*

*Creating database applications depending upon the requirement of the user.*

#### 6. Elementary Concept of Objects and Classes

Modelling entities and their behaviour by objects; a class as a specification for objects and as an object factory; computation as message passing/function calls between objects (many examples should be done to illustrate this). Objects encapsulate state (attributes) and have behaviour (functions). Class as a user defined type.

*A class may be regarded as a blueprint to create objects. It may be viewed as a factory that produces similar objects. A class may also be considered as a new data type created by the user, that has its own functionality.*

*All the four features of Object Oriented Programming should be defined and explained using real life examples.*

*Analyze each object and show how each contains attributes and responds to certain messages or permits certain operations.*

*Emphasize that an object is an instance of a class. A single object is just a bundle of values, one for each attribute in the class.*

#### 7. Values and types

Tokens and its types, Primitive types, operations on primitive values, expressions, assignment (assignment is also an expression).

*Introduce the primitive types and the range of values each represents. Discuss all the operations that can be done with primitive types namely mathematical, relational and logical. –Discuss precedence and associativity of operators. Introduce the concept of type casting.*

*Introduce System.out.println and System.out.print, for simple output.*

*Discuss different types of errors occurring during execution and compilation of the program (syntax errors, runtime errors and logical errors).*

### **8. Conditionals and non-nested loops**

Application of if else, if else if ladder, switch-case (default, break).

Fixed number of iterations- the for loop. Unknown number of iterations - while loop, do-while loop.

The conditional/ternary operator ( ? : ) should be introduced at this point.

Loops are fundamental to computation and their need should be shown by examples. -.

Examples: various number based problems: prime numbers, composite numbers, perfect numbers, fibonacci numbers, etc.

### **INTERNAL ASSESSMENT - 100 Marks**

This segment of the syllabus is totally practice oriented. The accent is on acquiring basic usage skills quickly and efficiently.

#### ***Assignments and Application Building***

Students should complete a number of laboratory assignments during the whole year to reinforce the concepts studied in the class.

*The students should build one real life application using elements from most of the packages used (topics 1-7). Great care should be exercised to ensure that most of the components of the packages are used while building the application.*

**Important:** In Class IX mostly primitive types should be used to construct Objects.

#### **Suggested list of Assignments:**

##### **For topics (1 – 5)**

1. Connect to net. If you are already connected start a browser and a search engine and perform a simple search on any of the subjects. Then perform a Boolean search for the same topic and note the difference. Create your own email ID and send an e-mail.
2. A story is given in which the main character is a lady, Mrs. James. Make changes in the document using find and replace so that the main character becomes a gentleman Mr. Brown. Take care of the pronouns.

3. Design a presentation to launch an awareness program like “Save the Earth”.
4. Create a presentation to highlight the activities in your school.
5. Build a friendly database to access phone nos., emails and addresses conveniently.

##### **For topics (6-8)**

The laboratory assignments will form the bulk of the course. Good assignments should have problems which require design, invention of an algorithm and only then implementation and testing. The problems will mimic a real life problem and require careful design or will require an interesting algorithm to solve it. They should also embody one or more concepts that have been discussed in the theory class. A significant proportion of the time has to be spent in the laboratory. Computing can only be learnt by doing.

Some sample problems are given below as examples. The problems are of varying levels of difficulty:

6. Enter marks of ten students in six subjects, find the total marks and the average of each student using Mathematical functions and make a chart on the basis of their average marks.
7. Create a table in a Word Processor for your class with the following fields: Roll No., Name, Date of Birth and Optional Subject.
8. Implement a Calculator class that models a hand held calculator. It should have (at least) the following functionality: addition, subtraction, multiplication, integer division, remainder, unary minus, enter, clear.
9. A student has a name, roll number, class in which studying, and marks in 6 subjects. Design a class for student. Write constructors, *get* and *set* functions and separate functions to return the total, percentage and grade.
10. Fibonacci series is obtained by adding the previous two terms. For example: 1,1,2,3,5,8,13,21...etc.

Fibonacci primes are prime numbers that belong to the Fibonacci series. For example 2,3,5,13... etc. Define a class NumberSeries with suitable

methods which takes an integer argument and prints out all the Fibonacci primes below that number.

11.  $n$  is a perfect number if the sum of all the factors of the number (including 1) excluding itself is  $n$ . For example:

$$6 = 1+2+3$$

$$28=1+2+4+7+14$$

Define a class called Number Problems which have the following functions:

*int sumOfFactors(int n)* - which returns sum of all the factors of the number  $n$  except itself.

*boolean isPerfect(int n)* – which returns *true* if the number  $n$  is perfect and *false* otherwise.

*void PerfectNosBelow(int lim)* - which first prints out all perfect numbers less than  $lim$ . Each perfect number should be printed on a single line along with its factors (see below). So for example the output from *PerfectNosBelow(10)* will be:

$$6 = (1,2,3)$$

**Important:** This list is indicative only. Teachers and students should use their imagination to create innovative and original assignments.

### Some Ideas for Application Building:

1. Trace the evolution of Computing over the centuries.
2. How computers affect human beings. Focus upon the Ergonomic, Psychological and Social aspects.
3. Visualize the future based on current developments in technology. Focus upon its impact and the ethical questions involved.
4. Find out how each component of the computer works. Build a presentation to explain what really goes on, inside a computer.
5. Build a dictionary or a thesaurus database and use it to find synonyms, antonyms and the pronunciation of words.
6. Build a project to store the data of various cricket teams and output it in a variety of ways. Make future projections on the basis of this data.
7. Collect the data on the state of air/water/noise pollution in your area with the help of your chemistry teacher. Use your computer to keep

track of this data and analyze it. Launch an awareness campaign and find ways to improve the situation. You can make pamphlets and presentations with the help of your computer.

8. Ask your teacher/ father/ mother/ Principal as to what are the documents they need for their work. Prepare a list and see if you can make some templates to help them. Help them with their budgets, accounting and future planning.
9. Write a small book of poems or quotations and index it. Build the table of contents.
10. You are building your own web page. What material will you require? Create it.
11. Visit your school library. Build a project to handle the issuing and returning of books.
12. Build a project to generate various quizzes and puzzles automatically.
13. Imagine you are organizing an event; say a three-day computer festival in your school. Use your computer to manage all the planning and details.
14. Visit your Bank. Find out about the various kinds of accounts e.g. Saving Accounts, Current Accounts, Fixed Deposit Accounts, and Recurring Deposit Accounts. Design a project to handle at least one kind of account.
15. Visit a small retail organization. Build a project to handle the bills and inventory.
16. Visit your school office. Study the teachers' payroll system. Design a system to print pay slips.
17. Visit a hotel. Study the various processes. Design a project to handle the booking and checking out.
18. Visit the LIC office. Study how they handle various policies. Design a project to handle any one kind of policy.
19. Design a Booklet (Giving details of the organization, courses offered and the additional facilities provide by the Institute), Handbills and Posters for advertisement for a newly established Computer Institute.
20. Create a multi page brochure for a travelling agency, including the following points:
  - Description of the places to be visited

- Schedules of various trips
- Costs Involved

**Important:** This list is indicative only. Teachers and students should use their imagination to create innovative and original applications.

### Application Building Assignments (Class IX)

#### Proposed Guidelines for Marking

The teacher should use the criteria below to judge the internal work done. Basically, four criteria are being suggested: analysis, design, execution and effectiveness. 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 internal teacher based on his/her judgment. However, one possible way: 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?

Have suitable tools been chosen?

#### Design:

Is the choice of data structures proper?

Is the logic suitable for the problem?

How efficient is it?

#### Execution:

Does the application run correctly?

#### Effectiveness:

Does the application solve the problem effectively?

Criteria (mm -40)	Analysis (mm-10)	Design (mm-10)	Execution (mm-10)	Effectiveness (mm-10)
Excellent	10	10	10	10
Good	8	8	8	8
Fair	6	6	6	6
Poor	4	4	4	4

## EVALUATION

The teacher in-charge shall keep the record of all the assignments and evaluate them internally.

The teacher-in-charge shall evaluate one application built by the student. He/She shall ensure that most of the components of the syllabus have been used appropriately in the chosen application.

### SOFTWARE FOR CLASS IX:

There is a wide variety of software packages and operating systems available but software has to be chosen very carefully. Schools are expected to explore any suitable Operating System or Software Package, which is being used currently and is likely to be used in future, and choose it for covering topics 1-5 of the syllabus.

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.

For covering topics 6-8

- Any suitable Operating System can be used.

For teaching fundamental concepts of computing using object oriented approach, BlueJ environment (1.2 or higher version) compatible with JDK (1.3 or higher version) as the base or any other editor or IDE, compatible with JDK (1.3 or higher version) as the base may be used. Ensure that the latest versions of software are used.

## CLASS X

There will be **one** written paper of **two hours** duration carrying 100 marks and Internal Assessment of 100 marks.

The paper will be divided into two sections A and B.

**Section A** (Compulsory – 40 marks) will consist of compulsory short answer questions covering the entire syllabus.

**Section B** (60 marks) will consist of questions which will require detailed answers and there will be a choice of questions in this section

### THEORY – 100 Marks

#### 1. Revision of Class IX Syllabus

- (i) Elementary Concept of Objects and Classes.
- (ii) Values and types.
- (iii) Conditionals and non-nested loops.

#### 2. Class as the Basis of all Computation

Objects and Classes

Objects encapsulate state and behaviour – numerous examples; member variables; attributes or features. Variables define state; member functions; Operations/methods/ messages/ functions define behaviour.

Classes as abstractions for sets of objects; class as an object factory; concept of type, primitive data types, composite data types. Variable declarations for both types; difference between the two types. Objects as instances of a class.

Consider real life examples for explaining the concept of class and object.

#### 3. Constructors

Constructor and its types.

Default constructor, parameterized constructor, constructor with default parameter and constructor overloading.

#### 4. Functions

Functions and its types

Need of functions. Types of functions (pure and impure). Function declaration and definition, ways of calling functions (call by value and call by reference) Returning information/messages from the functions and use of multiple functions and more than one function with the same name (function overloading). Use of static data member with static member function. Discuss invocation of functions on objects (through the reference). Discuss the concept of this with a reference to the object on which the invocation is made again.

#### 5. Class as a User Defined Type

Class as a composite type, distinction between primitive type and composite or class types.

Class may be considered as a new data type created by the user, that has its own functionality.

The distinction between primitive and composite types should be discussed through examples. Show how classes allow user defined types in programs. All primitive types have corresponding class wrappers. The following methods are to be covered:

`int parseInt(String s)`, `int valueOf(String s)`,  
`long parseLong(String s)`, `long valueOf(String s)`,  
`float parseFloat(String s)`, `float valueOf(String s)`,  
`double parseDouble(String s)`,  
`double valueOf(String s)`, `boolean isDigit(char ch)`,  
`boolean isLetter(char ch)`,  
`boolean isLetterOrDigit(char ch)`,  
`boolean isLowerCase(char ch)`,  
`boolean isUpperCase(char ch)`,  
`boolean isWhitespace(char ch)`,  
`char toLowerCase(char ch)`  
`char toUpperCase(char ch)`

#### 6. Iterations.

Loops, nested loops, break and continue.

Revision of loops (while, do while and for).

Show how each kind of loop can be converted to the other form of the loop. Introduce nested loops through some simple examples. Demonstrate break and continue statements with the help of loops/nested loops.

#### 7. Using Library Classes

Simple input/output. String, packages and import statements.

*Browsing the documentation for classes in the libraries and illustrating their use. The following functions have to be covered:*

String library functions:  
Char charAt (unt i)  
int compareTo(String1, String2)  
String concat(String str)  
boolean endsWith(String str)  
boolean equals(String str)  
boolean equalsIgnoreCase(String str)  
int indexOf(char ch)  
int lastIndexOf(char ch)  
int length( )  
String replace (char oldChar,char newChar)  
boolean startsWith(String str)  
String substring(int beginIndex, int endIndex)  
String toLowerCase()  
String toUpperCase()  
String trim( )  
String valueOf(all types)  
Mathematical Library Functions:  
pow(x,y), log(x), sqrt(x), ceil(x), floor(x), rint(x),  
abs(a), max(a, b), min(a,b), random( ), sin(x), cos(x),  
tan(x).

*Introduce the concept of packages and import statement (Avoid discussing the details of libraries).*

## 8. Encapsulation

Access specifiers and scope and visibility

*Access specifiers – private and public. Visibility rules for private, package and public access specifiers. Scope of variables, instance variables, argument variables, local variables.*

## 9. Arrays

*Arrays –storing, retrieving and arranging data*

*Arrays and their uses, sorting algorithms - selection sort and bubble sort; Search algorithms – linear search and binary search —Example of a composite type. Array creation. Sorting and searching algorithms should be discussed (single dimensional array only).*

## 10. Input/Output

Basic input/output using Scanner and Printer classes from JDK.

*The Scanner class can be used for input of various types of data (e.g. int, float, char etc.) from the standard input stream.*

### INTERNAL ASSESSMENT - 100 Marks

#### Assignments and Project

The students should complete a number of laboratory assignments during the whole year to reinforce the concepts studied in the class.

The students should build one real life project using the concepts taught.

#### Suggested list of Assignments:

Good assignments should have problems which require design, invention of an algorithm and only then implementation and testing. The problems will mimic a real life problem and require careful design or will require an interesting algorithm to solve it. They should also embody one or more concepts that have been discussed in the theory class. A significant proportion of the time has to be spent in the laboratory. Computing can only be learnt by doing. Some sample problems are given below as examples. The problems are of varying levels of difficulty.

1. A student has a name, roll number, class in which studying, home address and a date of birth. Design a class containing – constructors and user define functions, *get* and *set* – *get* to input data and *set* to display data. - . - .
2. Write a class *Convert* with methods as follows:
  - a) takes 4 arguments representing miles, yards, feet and inches and convert them into kilometres, meters and centimetres.
  - b) takes an argument representing degrees Fahrenheit and convert it to degrees centigrade.
  - c) a kilobyte is interpreted in two ways: some times it is 1000 bytes (actually correct), but often (and traditionally) it is  $2^{10}$  which is 1024. Similar discrepancies arise for mega, giga, tera and peta (each is 1000 (or  $2^{10}$ ) times the previous one).

The function should take the  $10^3$  (standard kilo) and give the equivalent value using  $2^{10}$  as a kilo for all the above.

3. Define a class `RecurringPatterns` and define methods in it which will print the following patterns.

a) The method takes an integer argument `n` and prints the following pattern, shown for `n=4`.

```

a
a a
a a a
a a a a
a a a
a a
a

```

b) The method takes an integer argument and prints the following pattern, shown for `n=4`.

```

1
121
12321
1234321
12321
121
1

```

c) The method takes an integer argument and prints the following pattern, shown for `n=4`.

```

abcdcba
abc cba
ab ba
a a
ab ba
abc cba
abcdcba

```

**Note:** for the three methods above you can assume that `n < 10`. However, think about what you would do if you allowed `n` to be a 2 or even 3 digit integer.

4. Define class `Point` to model points in the X-Y plane. Define functions to translate a point along the X and Y axes respectively. Define a function that calculates the distance from another point.

5. Write a program to input two strings. Check both the strings and remove all common characters from both the strings. Print both the strings after removing the common characters.

**Important:** This list is indicative only. The teachers and students should use their imagination to create innovative and original assignments.

**Some Ideas for the Project:**

*Students have already been introduced to spreadsheets, databases, word processors and presentation software earlier. That familiarity should be used to introduce the idea of how the software can be designed by modeling it as operations permitted on different objects. Other real world systems can also be modeled on the same lines:*

1. Calculators
2. Banks
3. A school
4. Toys
5. A game
6. Traffic lights
7. Elevators
8. Retail Outlets
9. An office
10. A company
11. Household gadgets like microwave ovens
12. Washing machines
13. Air Conditioners
14. Cars
15. Airplanes
16. Vending machines
17. Automatic Teller Machines (ATM )
18. A Social System
19. A musical composition
20. A clinical diagnostic system

**Important:** This list is indicative only. The teachers and students should use their imagination to create innovative and original projects.

**Programming Project (Class X)**

## Proposed Guidelines for Marking

The teacher should use the criteria below to judge the internal work done. Basically, four criteria are being suggested: class design, 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 possible way: 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.

### Class design:

Has a suitable class (or classes) been used?

### Execution:

Does the program run on all sample input correctly?

Criteria (Total – 40m)	Class Design (mm-10)	Algorithm Design (mm-10)	Coding and Documentation (mm-10)	Execution (mm- 10)
Excellent	10	10	10	10
Good	8	8	8	8
Fair	6	6	6	6
Poor	4	4	4	4

## EVALUATION:

The teacher in-charge shall keep the record of all the assignments done by the student throughout the year and evaluate them internally. The student is expected to do one project under the guidance of the teacher in charge.

An External Examiner shall be nominated by the Head of the school 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 Applications Projects.

Evaluation of practical work will be done as follows:

Subject Teacher (Internal Examiner) : 50 marks

External Examiner : 50 marks

Are all attributes with the right kinds of types present?

Is encapsulation properly done?

Is the interface properly designed?

### 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? (class documentation, variable documentation, function documentation, constraints, known bugs - if any)

The total marks obtained out of 100 are to be sent to the Council by the Head 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 course shall require at least 4 periods of about 40 minutes duration per week. In one week out of 4 periods the time should be divided as follows:

- 1 period – Lecture cum demonstration by the Instructor
- 3 periods – Assignments and Project

*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 minimal specifications as of now.*

#### **RECOMMENDED 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 at least a PIII or Equivalent Processor.***
- Good Quality printers.
- A scanner, a web cam/a digital camera (Should be provided if possible).

#### **SOFTWARE FOR CLASS X**

*Any suitable Operating System can be used.*

For teaching fundamental concepts of computing using object oriented approach, BlueJ environment (1.2 or higher version) compatible with JDK (1.3 or higher version) as the base or any other editor or IDE, compatible with JDK (1.3 or higher version) as the base may be used. Ensure that the latest versions of software are used.