

## COMPUTER APPLICATIONS (86)

### The Aim:

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 & 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) Command User Interface

The need for an Operating System, the Booting Process, Directory handling, Absolute and Relative path names, File handling, Disk handling Commands, Batch Files.

##### ii) Graphic User Interface

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

##### iii) Installing the Operating System

Installing the Operating System and other software, setting up printers.

#### 2. The Internet

Communicating through Computers, creating an e-mail I.D., e-mail and ftp, POP and IMAP and SMTP for mail and the ftp protocol for file transfer; using the World Wide Web, HTTP and web servers; downloading information; using Search engines, configuring the Internet.

#### 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.

#### 4. Word Processing

Starting and closing the word processor under a graphic user interface; creating and saving a document; protecting a document; editing a document, find and replace, page layout, printing a document, formatting a document, enhancing a document, proofing a document, mail merge; labels & envelopes; creating indexes and table of contents. Templates, embedding objects.

#### 5. Multimedia Presentation

The art and science of communication. Planning a presentation; creating and saving a presentation in different formats; slide layout and transition; working with graphics, embedding audio and video clips; recording dialogue and sounds; embedding scanned images and objects from other software. Integrating

the presentation with word processors and the internet, animation and design enhancement.

## 6. Spreadsheets

The need for spreadsheets; creating and saving a spreadsheet; working with formulae; relative and absolute referencing; editing a spreadsheet, using functions, working with ranges, graphs and charts; formatting a spreadsheet; working with multiple spreadsheets; printing spreadsheets; importing external data and working with it. Tools for analysis.

## 7. 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; mailing labels; report generation. Working with multiple databases, object linking and embedding, creating applications.

## 8. 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.

## 9. Values and types

Primitive types (like int, float, boolean etc.) and their representation and ranges, operations on primitive values, expressions, assignment (assignment is also an expression), attributes, constructors, functions. Return values and function parameters, passing parameters by value.

## 10. Conditionals and non-nested loops

if-then, if-then-else.

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

## 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 (1-7):

1. Make a complex directory structure. Switch from one directory to another using Absolute path and Relative path. Copy the contents of one directory to another.
2. Create files at different locations. Copy and move a group of files from one location to another using wild card characters. Change file attributes and then work with them. After all the work has been done, delete the file and then undelete it and see the changes.
3. 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.
4. 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.
5. Design personalized invitations for a party by using Mail Merge.

6. Design a presentation to launch an awareness program like “Save the Earth”.
7. Create a presentation to highlight the activities in your school.
8. Make projections of the chances of winning or loosing a game on the basis of the past performance of a team.
9. Do a ‘What if Analysis’ for setting up a new toy factory.
10. Build a friend database to access phone nos., emails and addresses conveniently.

### **For topics (8-10):**

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:

11. 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.
12. 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.
13. 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.

14.  $n$  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 perfectNos Below(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 *perfect Nos Below(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 1-7 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 8-10;

- 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 & 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 encapsulate state and behavior – numerous examples; member variables; attributes or features.

Variables define state; member functions; Operations/methods/ messages/ functions define behavior.

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. Modeling by composition.

#### 3. Constructors

Default constructor, constructors with arguments, default initialization. Overloading constructors.

#### 4. Functions

Function as a way to define operations/methods/ messages; pure functions return values and do not change state; impure functions may return values but also change state, return types arguments to functions, function prototype and function signature, overloading. Variables of class type as references to an object, invocation of functions on objects through the reference, the concept of this. Argument passing

in functions, pass by value, what happens when a reference is passed - side effects.

### 5. Class as a User Defined Type

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

### 6. Decision Making

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

### 7. Iterations.

Utilization of loops. Fixed number of iterations. The for loop, Unknown number of iterations - while loop, do-while loop, (for loop can also be used but should be used mainly when number of iterations are fixed), continue, break. Nested loops.

### 8. Using Library Classes

Simple input, output. String, static variables and static methods, packages and import statements.

### 9. Encapsulation

Private, public, scope and visibility rules. ; packages and package level access.

### 10. Arrays

Arrays and their usage; sorting algorithms - selection sort and bubble sort; Search in sorted arrays. The class Object - compatible with all class types.

### 11. Operations on Files

Streams - byte and character streams, files and operations on files, tokens and String Tokenizer and Stream Tokenizer classes.

**Important:** The teachers may write classes for I/O and give them to the students to use. The students may need Operations on Files for their projects. *However no questions shall be asked in the theory paper from Operations on Files.*

## 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. First design a suitable class for Date. Write constructors and *get* and *set* functions. Then design a class for student. Write constructors, *get* and *set* functions and a function to calculate the age of a student (use today's date in the function). The age should be returned as a triple (years, month, days). You will have to define a class for Triple and return an object of this type as the age.
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. Older computers used Binary Coded Decimals (BCD) to represent integers. In this each digit is represented by using 4 bits. So 0 is 0000, 1 is 0001, 2 is 0010, ... 9 is 1001. The remaining 4 bit patterns, namely 1010 onwards are illegal. So 32 bits can store up to 8 digits. To represent a negative number the sign is also encoded using 4 bits. However, we are interested only in positive BCD numbers. For example assuming we have 32 bits the number 123 will be represented as (0000) 5 times followed by 000100100011. Define a class *BCD* with a single constructor, which takes an *integer* argument and creates its BCD equivalent. Write methods for adding, subtracting BCD numbers.

4. 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.

5. 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.

6. Now use the `Point` class to define a `Rectangle` class where a rectangle is modelled by specifying all the four corners of the rectangle. Write separate functions to:

- Calculate the perimeter and area of the rectangle.
- Translate the rectangle along the X-axis, Y-axis and X-Y axes (simultaneously).

7. We want to build the basic backbone of a simple railway reservation system. The following classes are required to model the system: `Person`, `Date`, `Train`, and `Rail Reservation Record`. You have to invent the requisite attributes and functions for the various classes based on a description of the functionality and constraints of `Rail Reservation Record` object given below:

- Reservation is made from a start station to a destination station on a single train, in a particular class, for a particular date, for a person. So we are not taking care of journeys requiring two or more trains.
- Reservation can be made up to a maximum of 60 days and up to a minimum of 1 day before the date of departure.
- Reservations already made can be canceled.
- Assume that the trains have infinite capacity. So you need not make checks to see whether space is available or not.
- For reserving accommodation each passenger's name, gender and age is required.
- Each `Rail Reservation Record` object must have a unique reference number/string for access purposes.

You can use classes you may have defined in some of your earlier assignments. You can also invent more classes than the ones listed above if you think they are necessary. All your classes should have proper access declarators `private`, `public` etc.

**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?

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)

#### 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 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 Applications Projects.

Evaluation of practical work will be done as follows:

Subject Teacher (Internal Examiner) : 50 marks

External Examiner : 50 marks

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.