# 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 develop logical and analytical thinking so that they can easily solve interactive programs.
4. To help students learn fundamental concepts of computing using object oriented approach in one computer language.
5. 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. There will be a choice of questions in this section.*

### THEORY – 100 Marks

**1. Introduction to Object Oriented Programming concepts**

(i) Principles of Object Oriented Programming, (Difference between Procedure Oriented and Object oriented).
*All the four principles of Object Oriented Programming should be defined and explained using real life examples (Data abstraction, Inheritance, Polymorphism, Encapsulation).*

(ii) Introduction to JAVA - Types of java programs – Applets and Applications, Java Compilation process, Java Source code, Byte code, Object code, Java Virtual Machine (JVM), Features of JAVA.

*Definition of Java applets and Java applications with examples, steps involved in compilation process, definitions of source code, byte code, object code, JVM, features of JAVA - Simple, Robust, secured, object oriented, platform independent, etc.*

**2. 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/method calls between objects (many examples should be done to illustrate this). Objects encapsulate state (attributes) and have behaviour (methods). Class as a user defined data 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.*

**3. Values and data types**

Character set, ASCII code, Unicode, Escape sequences, Tokens, Constants and Variables, Data types, type conversions.

*Escape sequences [\n, \t, \\, \", \'], Tokens and its types [keywords, identifiers, literals, punctuators, operators], primitive types and non-primitive types with examples, Introduce the primitive types with size in bits and bytes, Implicit type conversion and Explicit type conversion.*

**4. Operators in Java**

Forms of operators, Types of operators, Counters, Accumulators, Hierarchy of operators, 'new' operator, dot ( . ) operator.

*Forms of operators (Unary, Binary, Ternary), types of operators (Arithmetic, Relational, Logical, Assignment, Increment, Decrement, Short hand operators), Discuss precedence and associativity of operators, prefix and postfix, Creation of dynamic*

*memory by using new operator, invoking members of class using dot operator, Introduce System.out.println() and System.out.print() – for simple output.*

***(Bitwise and shift operators are not included).***

**5. Input in Java**

Initialization, Parameter, introduction to packages, Input streams (Scanner Class), types of errors, types of comments

*Initialization – Data before execution, Parameters – at the time of execution, input stream – data entry during execution – using methods of Scanner class [nextShort(), nextInt( ), nextLong( ), nextFloat ( ), nextDouble( ), next( ), nextLine( ), next ( ) .charAt(0)* ]

*Discuss different types of errors occurring during execution and compilation of the program (syntax errors, runtime errors and logical errors). Single line comment (//) and multiline comment (/ * …. * / )*

**6. Mathematical Library Methods**

Introduction to package java.lang [ default ], methods of Math class.

*pow(x,y), sqrt(x), cbrt(x), ceil(x), floor(x), round (x), abs(a), max(a, b), min(a,b), random( ).*

*Java expressions – using all the operators and methods of Math class.*

**7. Conditional constructs in Java**

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

*if, if else, if else if, Nested if, switch case, break statement, fall through condition in switch case, Menu driven programs, System.exit(0) - to terminate the program.*

**8. Iterative constructs in Java**

Definition, Types of looping statements, entry controlled loops [ for, while], exit controlled loop [do while] , variations in looping statements, and Jump statements.

*Syntax of entry and exit controlled loops, break and continue, Simple programs illustrating all three loops, inter conversion from for – while – do while, finite and infinite, delay, multiple counter variables (initializations and updations). Demonstrate break and continue statements with the help of loops.*

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

**9. Nested for loops**

Introduce nested loops through some simple examples. Demonstrate break and continue statements with the help of nested loops.

*Programs based on nested loops [ rectangular, triangular [right angled triangle only] patterns], series involving single variable*

***(Nested while and nested do while are not included).***

**10. 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, cybercrime, hacking, protection against malicious intent and malicious code. The stress should be on good etiquette and ethical practices.*

**INTERNAL ASSESSMENT - 100 Marks**

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

**Programming Assignments (Class IX)**

Students are expected to do a minimum of 20 assignments during the whole year to reinforce the concepts studied in the class.

**Suggested list of Assignments:**

The laboratory assignments will form the bulk of the course. Good assignments should have problems which require design, implementation and testing. 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.

The teacher-in-charge should maintain a record of all the assignments done as a part of practical work throughout the year and give it due credit at the time of cumulative evaluation at the end of the year.

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

(i) Programs using Assignment statements.
Example: Calculation of Area / Volume / Conversion of temperature / Swapping of values etc.

(ii) Programs based on– Input through parameters.
Example: Implementation of standard formula etc.

(iii) Programs based on – Input through Scanner class.
Example: Implementation of standard formula etc.

(iv) Programs based on Mathematical methods.
Example: larger/smaller of two numbers, cube root, square root, absolute value, power, etc.

(v) Programs based on if, if else, if else if ladder, nested if etc.

(a) if programs

- Larger / smaller of two numbers
- To check divisibility of a number, etc.

(b) if -else programs

- Odd or even number
- Eligibility to vote
- Upper case or lower case
- Positive or negative number
- Vowel or Consonant
- Buzz number etc.

(c) if-else-if programs

- Programs based on discount/interest/ bonus/ taxes/ commission.
- Programs based on slab system.
- Programs based on Nested if.

(vi) Programs on switch case.

(a) Day of a week
(b) Name of the month
(c) Names of the seasons
(d) Calculator
(e) Vowel or consonant etc.

(vii) Programs based on Looping Statement

(a) Programs based on for looping statement

(b) Programs based on printing simple series, summation of simple series, product of simple series.

(c) Prime number, perfect number, composite number, Fibonacci series. Lowest Common Multiple (LCM), Highest Common Factor (HCF) etc.

(d) To find the biggest and smallest number from *n* number of entered numbers

(e) Program based on while loop like Armstrong number, Spy number, Niven number, Palindrome number, etc.

(viii) Programs based on nested loops [rectangular, triangular(right angled triangle only) patterns], series involving single variable

(ix) Generate first *n* multiples of numbers from 1 to the limit input by the user.

(x) Menu Driven programs.

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

## EVALUATION

### 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, coding and documentation, variable description and execution or output. 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.

**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?

**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, method documentation, constraints, known bugs – if any).

**Variable and Description**

**Format for variable description:**

| Name of the variable | Data Type | Purpose/Description |
|---|---|---|
| | | |
| | | |

**Evaluation of practical work (Assignments) will be done as follows:**

Subject Teacher (Internal Examiner): 100 marks

| Criteria (Total-100 marks) | Class design (20 marks) | Variable description (20 marks) | Coding and Documentation (20 marks) | Execution OR Output (40 marks) |
|---|---|---|---|---|
| Excellent | 20 | 20 | 20 | 40 |
| Good | 16 | 16 | 16 | 32 |
| Fair | 12 | 12 | 12 | 24 |
| Poor | 8 | 8 | 8 | 16 |