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 to LowerCase (char ch)
char to UpperCase(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. 173

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

String library functions:
Char charAt (int n)
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. 174

3. Define a class Recurring Patterns 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.*

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 – 40 marks)	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