

Strictly based on the latest ICSE Curriculum

 **OSWAAL BOOKS**
LEARNING MADE SIMPLE

For
2019
Exam



ICSE



SOLVED PAPER 2018



CLASS 10

COMPUTER APPLICATIONS



 **OSWAAL BOOKS**

1/11, Sahitya Kunj, M.G. Road,

Agra - 282002, UP (India)

Ph.: 0562 2857671, 2527781

email: contact@oswaalbooks.com

website: www.oswaalbooks.com

0504

Disclaimer :

Oswaal Books has exercised due care and caution in collecting the data before publishing this book. In spite of this if any omission, inaccuracy or printing error occurs with regards to the data contained in this book, Oswaal books will not be held responsible or liable. Oswaal Books will be grateful if you could point out any such error or your suggestions which will be of great help for other readers.



LATEST SYLLABUS

Computer Applications

CLASS 10

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 (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()`



...contd.

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 *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.

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

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.



ICSE Solved Paper, 2018

Class-X

Computer Applications

(Maximum Marks : 100)

(Time allowed : Two Hours)

Answers to this Paper must be written on the paper provided separately.

You will **not** be allowed to write during the first 15 minutes.

This time is to be spent in reading the question paper.

The time given at the head of this Paper is the time allowed for writing the answers.

This Paper is divided into two Sections.

Attempt **all** questions from **Sections A** and **any four** questions from **Section B**.

The intended marks for questions or parts of questions are given in brackets [].

SECTION - A

(40 Marks)

Attempt **all** questions

1. (a) Define abstraction. 2
- (b) Differentiate between searching and sorting. 2
- (c) Write a difference between the functions `isUpperCase ( )` and `toUpperCase ( )`. 2
- (d) How are private members of a class different from public members ? 2
- (e) Classify the following as primitive or non- primitive data types : 2
 - (i) `char`
 - (ii) `arrays`
 - (iii) `int`
 - (iv) `classes`
2. (a) (i) `int res = 'A';` 2
What is the value of `res`?
 - (ii) Name the package that contains wrapper classes.
- (b) State the difference between **while** and **do while** loop. 2
- (c) `system.out.print ("BEST");` 2
`system.out.println("OF LUCK");`
Choose the correct option for the output of the above statements
 - (i) BEST OF LUCK
 - (ii) BEST
OF LUCK
- (d) Write the prototype of a function **check** which takes an integer as an argument and returns a character. 2
- (e) Write the return data type of the following function. 2
 - (i) `endsWith ( )`
 - (ii) `log ( )`
3. (a) Write a Java expression for the following : 2

$$\frac{\sqrt{3x+x^2}}{a+b}$$

- (b) What is the value of y after evaluating the expression given below ? 2
`y+ = ++y + y-- + --y;` when `int y=8`
- (c) Give the output of the following : 2
 (i) `Math.floor(-4.7)`
 (ii) `Math.ceil(3.4) + Math.pow(2, 3)`
- (d) Write two characteristics of a constructor. 2
- (e) Write the output for the following : 2
`system.out.println("Incredible"+"\\n"+"world");`
- (f) Convert the following **if else if** construct into **switch case** 2
`if( var==1)`
`System.out.println("good");`
`else if(var==2)`
`System.out.println("better");`
`else if(var==3)`
`System.out.println("best");`
`else`
`System.out.println("invalid");`
- (g) Give the output of the following string functions : 2
 (i) `"ACHIEVEMENT".replace('E', 'A')`
 (ii) `"DEDICATE".compareTo("DEVOTE")`
- (h) Consider the following String array and give the output 2
`string arr[] = {"DELHI", "CHENNAI", "MUMBAI", "LUCKNOW", "JAIPUR"};`
`system.out.println(arr[0].length()>arr[3].length());`
`system.out.print(arr[4].substring(0, 3));`
- (i) Rewrite the following using ternary operator : 2
`if (bill>10000)`
`discount = bill * 10.0/100;`
`else`
`discount = bill * 5.0/100;`
- (j) Give the output of the following program segment and also mention how many times the loop is executed : 2
`int i;`
`for (i = 5; i > 10; i++)`
`system.out.println(i);`
`system.out.println(i*4);`

SECTION - B

(60 Marks)

*Attempt **any four** questions from this Section*

*The answer in the Section should consist of the **Programs in either Blue J environment or any program environment with Java as the base.***

*Each program should be written using **Variable descriptions/Mnemonic Codes** so that the logic of the program is clearly depicted.*

*Flow-Charts and Algorithms **are not required.***

4. Design a class `RailwayTicket` with following description : 15
- | | | |
|---------------------------------|---|---|
| Instance variables/data members | : | |
| <code>String name</code> | : | To store the name of the customer |
| <code>String coach</code> | : | To store the type of coach customer wants to travel |
| <code>long mobno</code> | : | To store customer's mobile number |
| <code>int amt</code> | : | To store basic amount of ticket |

To know about more useful books for class-10 [click here](#)

int totalamt : To store the amount to be paid after updating the original amount

Member methods :

void accept () – To take input for name, coach, mobile number and amount.

void update() – To update the amount as per the coach selected
(extra amount to be added in the amount as follows)

Type of Coaches	Amount
First_AC	700
Second_AC	500
Third_AC	250
Sleeper	None

void display()–To display all details of a customer such as name, coach, total amount and mobile number.

Write a main method to create an object of the class and all the above member methods.

5. Write a program to input a number and check and print whether it is a **Pronic** number or not. (Pronic number is the number which is the product of two consecutive integers) **15**

Example : $12 = 3 \times 4$

$20 = 4 \times 5$

$42 = 6 \times 7$

6. Write a program in Java to accept a string in lower case and change the first letter of every word to upper case. Display the new string. **15**

Sample input : we are in cyber world

Sample output : We Are In Cyber World

7. Design a class to overload a function volume() as follows : **15**

(i) double volume (double R) – with radius (R) as an argument, returns the volume of sphere using the formula.

$$V = \frac{4}{3} \times \frac{22}{7} \times R^3$$

(ii) double volume (double H, double R) – with height(H) and radius(R) as the arguments, returns the volume of a cylinder using the formula.

$$V = \frac{22}{7} \times R^2 \times H$$

(iii) double volume (double L, double B, double H) – with length(L), breadth(B) and Height(H) as the arguments, returns the volume of a cuboid using the formula.

$$V = L \times B \times H$$

8. Write a menu driven program to display the pattern as per user's choice **15**

Pattern 1

ABCDE

ABCD

ABC

AB

A

Pattern 2

B

LL

UUU

EEEE

For an incorrect option, an appropriate error message should be displayed.

9. Write a program to accept name and total marks of N number of students in two single subscript array **name[]** and **totalmarks []**. **15**

Calculate and print :

(i) The average of the total marks obtained by N number of students.

$$[\text{average} = (\text{sum of total marks of all the students})/N]$$

(ii) Deviation of each student's total marks with the average.

$$[\text{deviation} = \text{total marks of a students} - \text{average}]$$



ANSWERS

SECTION - A

(40 Marks)

Attempt all questions

1. (a) Abstraction is one of the key concepts of object-oriented programming (OOP) languages. Its main goal is to handle complexity by hiding unnecessary details of the object created by user. It enables the user to implement more complex logic on top of the provided abstraction without understanding or even thinking about all the hidden complexity. For example the forward and backward movement of the stereo set is abstracted by buttons like <=> and =>.
- (b) Searching is the process of finding one or all locations of any element in the given list of data and deciding if the element is present in the list or not. If that element is present in the list of data then searching is said to be successful otherwise unsuccessful.
- Sorting is the process of ordering the given data list in ascending or descending manner (numerically, alphabetically or lexicographically).
- (c) isUpperCase() is an in-built function of Java, it is used to check if the letters of the given string is in either uppercase or not. It returns true if all letters are in uppercase otherwise returns false.
- toUpperCase() is also an implicit function of Java that is used to change the lowercase letters of the given string to uppercase.
- (d) Private members can be accessed only within the class where they are declared, not outside the class, but public members can be accessed inside as well as outside the class.
- (e) (i) char → primitive data type
 (ii) arrays → non-primitive data type
 (iii) int → primitive data type
 (iv) classes → non-primitive data type
2. (a) (i) Value of res is 65 because the ASCII value of character 'A' in decimal is 65
 (ii) java.lang
- (b)

S.No.	while	do-while
1.	while(condition) { Statements; }	do { Statements; } while(condition);
2.	while loop is entry control loop.	do-while is exit control loop.
3.	In while loop, firstly condition is checked. If condition is true, statements within while block are executed. After then again Condition is checked and so on.	In do-while loop firstly statements within do-while block are executed, after then condition is checked. If condition is true, again statements are executed and so on.
4.	In while loop minimum execution of statements is zero.	In do-while loop minimum execution of statements is one.

- (c) (ii) BEST
 OF LUCK

To know about more useful books for class-10 [click here](#)

- (d) char check (int x); char check (int x);
- (e) (i) boolean
(ii) double
3. (a) $\text{Math.sqrt}(3*x + x*x)/(a+b);$
- (b) The expression $y += ++y + y -- + --y$ can be written as $y = y + (++y + y-- + --y)$.
If initial value of y is 8 then final value of $y = 8 + (9 + 9 + 7) = 33$.
- (c) (i) -5.0
(ii) $4 + 8 = 12$
- (d) 1. Constructor has the same name as that of the class it belongs to.
2. Constructor is executed automatically when an object of class is created.
- (e) Incredible
world
- (f) `switch(var)`
{
case 1: System.out.println("good");
break;
case 2: System.out.println("better");
break;
case 3: System.out.println("best");
break;
default: System.out.println("invalid");
}
- (g) (i) ACHIAVAMANT
(ii) -18
- (h) `arr[0]` denotes to "DELHI" and `arr[3]` denotes to "LUCKNOW". "DELHI" length is not greater than "LUCKNOW" length.
`arr[4]` denotes to "JAIPUR", so substring of "JAIPUR" from 0 to 2 will be "JAI".
So Output will be:
false
JAI
- (i) `discount = (bill > 10000)? bill * 10.0/100 : bill * 5.0/100;`
- (j) **output will be :**
20
The loop will be executed zero times.
4. `import java.io.*;`
`class RailwayTicket`
{
String name, coach;
long mobno;
int amt, totalamt;
void accept()
{
BufferedReader br= new BufferedReader(new InputStreamReader(System.in));
System.out.println("Enter customer name");
name= br.readLine();
System.out.println("Enter type of coach from(First_AC, Second_AC, Third_AC, sleeper)");

```

coach= br.readLine();
System.out.println("Enter cutomer mobile number");
mobno= Long.parseLong(br.readLine());
System.out.println("Enter basic amount of ticket");
amt= Integer.parseInt(br.readLine());
    }
void update()
    {
if(coach.compareTo("First_AC")==0)
totalamt= amt+ 700;
else if(coach.compareTo("Second_AC")==0)
totalamt= amt+ 500;
else if(coach.compareTo("Third_AC")==0)
totalamt= amt+ 250;
else if(coach.compareTo("sleeper")==0)
totalamt=amt;
    }
void display()
    {
System.out.println("Customer name- "+name);
System.out.println("Coach type- "+coach);
System.out.println("Total amount- "+totalamt);
System.out.println("Mobile Number- "+mobno);
    }
public static void main(String args[])
    {
RailwayTicket ob1= new RailwayTicket();
ob1.accept();
ob1.update();
ob1.display();
    }
}

```

```

5. import java.io.*;
class PronicNum
{
public static void main(string args[])
    {
BufferedReader br= new BufferedReader(new InputStreamReader(System.in));
System.out.println("Enter one number ");
int n= Integer.parseInt(br.readLine());
int i;
for(i=1; i*(i+1)<=n ; i++)
    {
if(n==i*(i+1))
    {
System.out.println(n+" is Pronic Number");
break;

```

To know about more useful books for class-10 [click here](#)

```

    }
    }
    System.out.println(n+ " is not Pronic Number");
    }
6. import java.io.*;
   class ChangeLetter
   {
   public static void main(string args[])
   {
   BufferedReader br= new BufferedReader(new InputStreamReader(System.in));
   String s,t="";
   System.out.println("Enter one string in lowercase");
   s=br.readLine();
   int i;
   char ch=s.charAt(0);
   ch=Character.toUpperCase(ch);
   t=t+ch;
   for(i=1;i<s.length();i++)
   {
   ch= s.charAt(i);
   if(Character.isWhitespace(ch)== true)
   {
   t=t+ch;
   ch=Character.toUpperCase(s.charAt(i+1));
   i++;
   }
   t=t+ch;
   }
   s=t;
   system.out.println(s);
   }
}
7. class class_volume
   {
   double volume(double R)
   {
   double V;
   V= (4.0/3)*(22.0/7)*Math.pow(R,3);
   return(V);
   }
   double volume(double H, double R)
   {
   double V;
   V= (22.0/7)* R * R * H;
   return(V);
   }
   double volume(double L, double B, double H)

```

To know about more useful books for class-10 [click here](#)

```
{
double V;
    V= L * B * H;
return(V);
}
}
```

8. `import java.io.*;`
`class pattern`
`{`
`public static void main(string args[])`
`{`
`BufferedReader br= new BufferedReader(new InputStreamReader(System.in));`
`System.out.println("Enter 1 for pattern1 and 2 for pattern2");`
`int choice= Integer.parseInt(br.readLine());`
`switch(choice)`
`{`
`case 1:`
`{`
`char i, j;`
`for(i='E';i>='A';i--)`
`{`
`for(j='A'; j<=i; j++)`
`System.out.print(j);`
`System.out.println();`
`}`
`break;`
`}`
`case 2:`
`{`
`String s="BLUE";`
`int m,n;`
`for(m=1;m<=4;m++)`
`{`
`charch=s.charAt(m-1);`
`for(n=1;n<=m;n++)`
`System.out.print(ch);`
`System.out.println();`
`}`
`break;`
`}`
`default: System.out.println("Invalid Input");`
`}`
`}`
`}`

9. `import java.io.*;`
`class Marks`
`{`

To know about more useful books for class-10 [click here](#)

```
public static void main(String args[])
{
    BufferedReader br= new BufferedReader(new InputStreamReader(System.in));
    System.out.println("Enter number of Students");
    int N= Integer.parseInt(br.readLine());
    String name[]= new String[N];
    int totalmarks[]= new int[N];
    for(int i=0;i<N;i++)
    {
        System.out.println("Enter student name");
        name[i]= br.readLine();
        System.out.println("Enter total marks of student");
        totalmarks[i]= Integer.parseInt(br.readLine());
    }
    double sum=totalmarks[0];
    for(int i=1;i<N;i++)
    {
        sum=sum+totalmarks[i];
    }
    double average= sum/N;
    System.out.println("Average of the total marks obtained by "+N+" Students is "+average);
    double deviation;
    for(int i=0;i<N;i++)
    {
        deviation= totalmarks[i]-average;
        System.out.println("Deviation of number "+(i+1)+" student is "+deviation);
    }
}
```

□□