

Strictly according to the Syllabus prepared by the Council for the Indian School Certificate Examinations,  
New Delhi for students appearing for Class X examination in 2021.

**UNDERSTANDING**

# **COMPUTER APPLICATIONS**

**with BlueJ**

**ICSE  
Class X**

**Vijay Kumar Pandey**

MCA, BCA, Diploma in Computers  
H.O.D., Computer Science  
Rajendra Vidyalaya, Sakchi  
Jamshedpur-831001 (Jharkhand)

**Dilip Kumar Dey**

B.Sc., B.Ed., Diploma in Computer Science  
Faculty, Computer Science  
Rajendra Vidyalaya, Sakchi  
Jamshedpur-831001 (Jharkhand)



**ARYA PUBLISHING COMPANY**

Village Johron, Trilokpur Road, Kala Amb 173 030, Distt. Sirmour (HP)

**Delhi Office:** 1002 Faiz Road (opp. Hanumanji Murti), Karol Bagh, New Delhi 110 005

The book has been published in good faith that the views and opinions expressed in this book are solely those of the original contributor(s)/authors.

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means [graphic, electronic or mechanical, including photocopying, recording, taping or information retrieval system] without the written permission of the copyright holder, application for which should be addressed to the publisher. Such written permission must also be obtained before any part of this publication is stored in a retrieval system of any nature. Breach of this condition is liable for legal action.

Exhaustive efforts have been made to ensure accuracy and correctness of contents of the book at the time of going to press. However, in view of possibility of human error or changes in the subject, neither the author, publisher nor any other person who has been involved in preparation of this work accepts any responsibility for any errors or omissions or results obtained from use of information given in the book.

The publisher shall not be liable for any direct, consequential, or incidental damages arising out of the use of the book.

In case of binding mistake, misprints, or missing pages etc, the publisher's entire liability, and your exclusive remedy, is replacement of the book within one month of purchase by similar edition/reprint of the book. In case of any dispute, all legal matters are to be settled under Delhi Jurisdiction only.

*Published by*

**ARYA PUBLISHING COMPANY**

Village Johron, Trilokpur Road

Kala Amb 173 030, Distt. Sirmour (HP)

*Delhi Office:*

1002 Faiz Road (opp. Hanumanji Murti),

Karol Bagh, New Delhi 110 005 (India)

Phone: 011-28752745, 28752604

Fax: 011-28756921

Email: [info@apcbooks.co.in](mailto:info@apcbooks.co.in)

Website: [www.apcbooks.co.in](http://www.apcbooks.co.in)

© Authors

ISBN-978-81-8296-688-8

First Edition : 2006

Second Edition : 2007

Reprint : 2008

Third Edition : 2009

Fourth Edition : 2010

Reprint : 2011

Fifth Edition : 2012

Modified Edition : 2013

Reprint : 2014

Sixth Edition : 2015

Reprint : 2016

Seventh Edition : 2017, 18

Revised Eighth Edition : 2019

Reprint : 2020

Price : ₹ 525.00

*Laser Typeset at:*

[Laser Tech Prints](#)

*Printed at:*

[Deepak Offset Printers](#)

Bawana Industrial Area, Delhi

## FOREWORD

It gives me a great pleasure in presenting my opening remarks for the book **UNDERSTANDING COMPUTER APPLICATIONS WITH BLUEJ** by Vijay Kumar Pandey and Dilip Kumar Dey (Senior teachers, Rajendra Vidyalaya). The book is written as per the latest ICSE syllabus for the Indian School Certificate Examinations, New Delhi.

Mr. Vijay Kumar Pandey and Mr. Dilip Kumar Dey have excellent academic background. My association with them has been for over a decade. I find them very hard working, competent and popular among the students. They make the concepts very interesting to learn, understand and apply. The academic performance of the students in Computer Applications / Computer Science under their guidance has been outstanding.

I am sure that the book will be very useful to the students preparing for the ICSE Examinations.

I wish them every success in their new ventures.



(Dipak Das Gupta)

Former Principal

Rajendra Vidyalaya

Jamshedpur (Jharkhand)

APC BOOKS

## PREFACE

We are really grateful to the students and teachers who have immensely appreciated the previous editions of the book and encouraged us to bring out this new edition of [Understanding Computer Applications with BlueJ](#).

This book, in its eighth edition, has been thoroughly revised as per the latest ICSE curriculum. The main thrust of the book is to deal with [Object Oriented Programming](#) in Java with [BlueJ](#).

The first chapter recapitulates the entire syllabus of Class IX in concise form. It is advised to refer 'Understanding Computer Applications with BlueJ', Class IX book for comprehensive understanding of the concepts.

The character and string manipulations have been designed separately along with the inclusion of Double Dimensional Array (DDA). Each chapter contains a number of review questions and exercises which are carefully designed to provide an insight into the subject. Strenuous efforts have been made to avoid errors specially programming errors. The programming examples have also been executed and verified on the computer. In spite of this, some errors may still be there. Students and readers are requested to bring the same to our knowledge.

Miscellaneous programs and project work for internal assessment have also been included in the book. Answers to objective type and very short answer type questions of each chapter have been given at the end of the book.

We express our gratitude to our family members and friends who always encouraged us to write this book. We are also thankful to our publishers who took pains in bringing out this book in such a short time.

Suggestions for further improvement will be gratefully acknowledged.

—Authors

# SYLLABUS

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. Revision of Class IX Syllabus

(i) Introduction to Object Oriented Programming concepts, (ii) Elementary Concept of Objects and Classes, (iii) Values and Data types, (iv) Operators in Java, (v) Input in Java, (vi) Mathematical Library Methods, (vii) Conditional constructs in Java, (viii) Iterative constructs in Java, (ix) Nested for 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 methods; Operations/methods/messages/methods define behaviour.

Classes as abstractions for sets of objects; class as an object factory; 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. User - defined Methods

Need of methods, syntax of methods, forms of methods, method definition, method calling, method overloading, declaration of methods.

Ways to define a method, ways to invoke the methods – call by value [with programs] and call by reference [only definition with an example], Object creation - invoking the methods with respect to use of multiple methods with different names to implement modular programming, using data members and member methods, Actual parameters and formal parameters, Declaration of methods – static and non-static, method prototype/signature, - Pure and impure methods, - pass by value [with programs] and pass by reference [only definition with an example], Returning values from the methods, use of multiple methods and more than one method with the same name (polymorphism - method overloading).

### 4. Constructors

Definition of Constructor, characteristics, types of constructors, use of constructors, constructor overloading.

Default constructor, parameterized constructor, constructor overloading, Difference between constructor and method

### 5. Library classes

Introduction to wrapper classes, methods of wrapper class and their usage with respect to numeric and character data types. Autoboxing and Unboxing in wrapper classes.

Class as a composite type, distinction between primitive data type and composite data type 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. Introduce Autoboxing and Unboxing with their definition and simple examples.

The following methods are to be covered:

`int parseInt(String s),`

`long parseLong(String s),`

*float parseFloat(String s),  
double parseDouble(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. Encapsulation

Access specifiers and its scope and visibility

*Access specifiers—private, protected and public. Visibility rules for private, protected and public access specifiers. Scope of variables, class variables, instance variables, argument variables, local variables.*

## 7. Arrays

Definition of an array, types of arrays, declaration, initialization and accepting data of single and double dimensional arrays, accessing the elements of single dimensional and double dimensional arrays.

*Arrays and their uses, sorting techniques—selection sort and bubble sort; Search techniques—linear search and binary search, Array as a composite type, length statement to find the size of the array (sorting and searching techniques using single dimensional array only).*

*Declaration, initialization, accepting data in a double dimensional array, sum of the elements in row, column and diagonal elements [right and left], display the elements of two-dimensional array in a matrix format.*

## 8. String handling

String class, methods of String class, implementation of String class methods, String array

*The following String class methods are to be covered:*

*String trim()  
String toLowerCase()  
String toUpperCase()  
int length()  
char charAt(int n)  
int indexOf(char ch)  
int lastIndexOf(char ch)  
String concat(String str)  
boolean equals(String str)  
boolean equalsIgnoreCase(String str)  
int compareTo(String str)  
int compareToIgnoreCase(String str)  
String replace(char oldChar, char newChar)  
String substring(int beginIndex)  
String substring(int beginIndex, int endIndex)  
boolean startsWith(String str)  
boolean endsWith(String str)  
String valueOf(all types)*

*Programs based on the above methods, extracting and modifying characters of a string, alphabetical order of the strings in an array [Bubble and Selection sort techniques], searching for a string using linear search technique.*

## CONTENTS

<b>1.</b>	<b>Revision of Class IX Syllabus</b>	<b>1</b>
	Unit I : Introduction to Object Oriented Programming Concepts	1
	Unit II : Elementary Concept of Objects and Classes	9
	Unit III : Values and Data Types	14
	Unit IV : Operators in Java	28
	Unit V (a) : Introduction to Java	39
	Unit V (b) : Input in Java	48
	Unit VI : Mathematical Library Methods	60
	Unit VII : Conditional Statements in Java	74
	Unit VIII : Iterative Constructs in Java	95
	Unit IX : Nested Loop	118
<b>2.</b>	<b>Library Classes</b>	<b>138</b>
<b>3.</b>	<b>Arrays (Single Dimensional and Double Dimensional)</b>	<b>164</b>
<b>4.</b>	<b>String Handling</b>	<b>224</b>
<b>5.</b>	<b>User – defined Methods</b>	<b>271</b>
<b>6.</b>	<b>Class as the Basis of all Computation (Objects and Classes)</b>	<b>316</b>
<b>7.</b>	<b>Constructors</b>	<b>361</b>
<b>8.</b>	<b>Encapsulation and Inheritance</b>	<b>392</b>
	Project Work	414
	Glossary	428
	Answers	433





## Revision of Class IX Syllabus

### Unit I: Introduction to Object Oriented Programming Concepts

#### LEARNING SCOPE

Introduction • Procedure-Oriented Language • Object Oriented Language • Principles of Object Oriented Programming (OOP) viz. Data abstraction, Encapsulation, Inheritance and Polymorphism.

#### Introduction

Language is a medium of communication. People can interact with each other if they know a common language. Similarly, a user can interact with a computer only if he/she knows a computer language. This is why we learn computer languages.

Computer languages are broadly classified into two types viz. low-level languages and high-level languages. As you know that low-level languages have certain limitations, therefore high-level languages have been developed. They provide a comfortable platform for the users to understand and design programming logic. The high-level languages are further classified into two types. They are:

- a. Procedure-Oriented
- b. Object Oriented



#### Procedure-Oriented Language

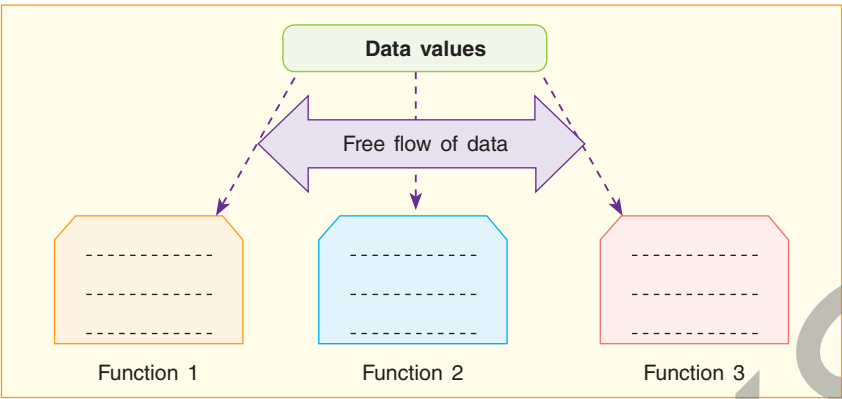
A programming language in which emphasis is given on the functions or procedures (actions to be taken) rather than data values is known as Procedure Oriented Programming language. It is a type of computer programming language that uses variables, functions, statements and conditional operators. It contains a systematic order of statements, procedures or functions and commands on the data values to complete a task. These procedures can be invoked anywhere between the program hierarchy and by other procedures as well.

It works basically on the following principles:

- Problems are divided into smaller programs called functions.
- The functions share global data.
- The data move freely from function to function.



It can be understood with the help of the following illustration:



*Procedure-Oriented Programming Concept*

The above logical structure illustrates that the data values have no restriction on their use in different functions of the program. They are independent to flow freely from one part of the program to other. Due to their free flowing nature, they may cause some complications in advanced or complex programming.

The limitations of procedure-oriented language are:

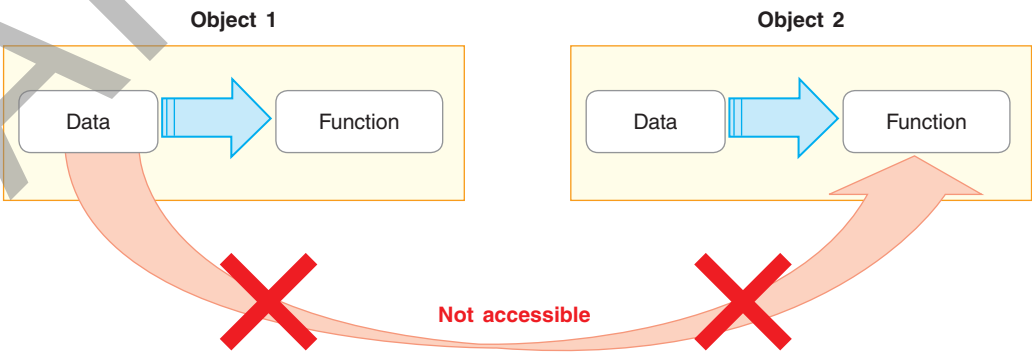
- No restriction on data values
- Cause problems for the programmers while debugging
- No reusability concept and hence time management, testing and length of the program increases.

Some high level languages such as BASIC, COBOL, FORTRAN, SL-1, PL-1, C, etc., are the procedure oriented languages.

‘C’ is a powerful computer language that is still used in engineering and scientific applications.

**Object Oriented Language**

To overcome the shortcomings of procedure oriented languages, the Object Oriented languages have been developed. In Object Oriented Programming, the stress is put on the data values rather than functions. The data values are confined and used within stipulated programming area. We can understand the concept with the help of the illustration given below:



*Object Oriented Programming Concept*

The above diagram illustrates that the object oriented programming allows user to split the complete program into a number of segments called 'Objects'. It is done in such a way that the data values of an object can only be used within the functions of the same object. They are not directly accessible to the other objects of the program. However, the objects can communicate with each other through functions.

Thus, object oriented programming is an approach in which stress is laid on data rather than functions. The data values remain associated with the functions of a particular block of the program so as to encourage data security.

The advantages of object oriented programming language are:

- Data values are secured.
- Mishandling of data is protected.
- Error detection and correction become easier.
- Easier in coding complex programs.

Java, C++, Small Talk, Eiffel, Python are some of the object oriented programming languages.

## Principles of Object Oriented Programming (OOP)

Once a program has been created by using different objects; the following principles will implicitly be embedded into it. They are:

### • Data Abstraction

In real life situation, you might have noticed that you need not require to know the details of the technologies to operate the system or a device.

For example, when you drive a car, you use steering wheel, accelerator, clutch and brake. You are not bothered about how they are designed to do their functions.

Similarly, in object oriented programming only the essential information about the data is known to the outside world. Hiding the background details or the implementation of code in order to reduce complexity is known as data abstraction.

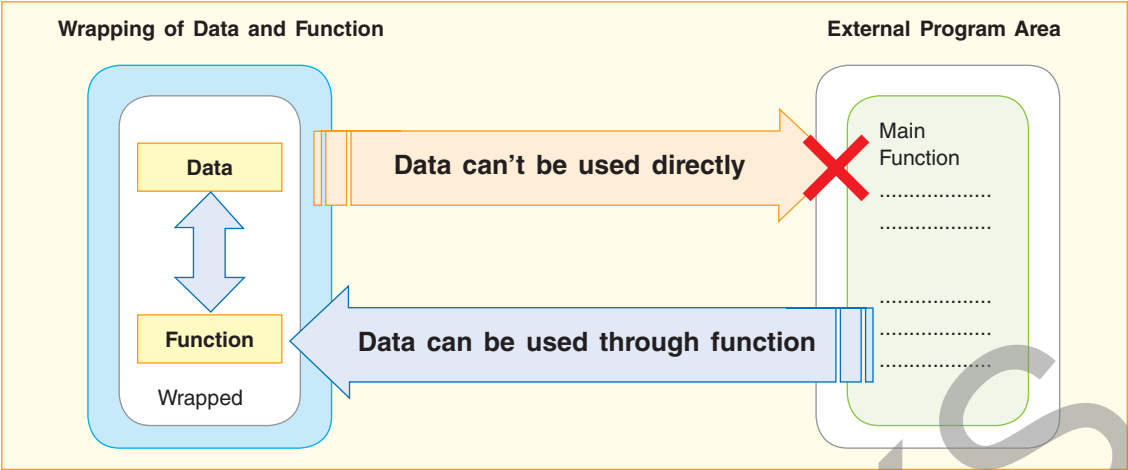


Thus, 'Data Abstraction' is defined as an act of representing the essential features without knowing the background details.

### • Encapsulation

In an object oriented programming, the data and functions are grouped or merged together in such a way that the data items are only accessible within the functions of the same object.





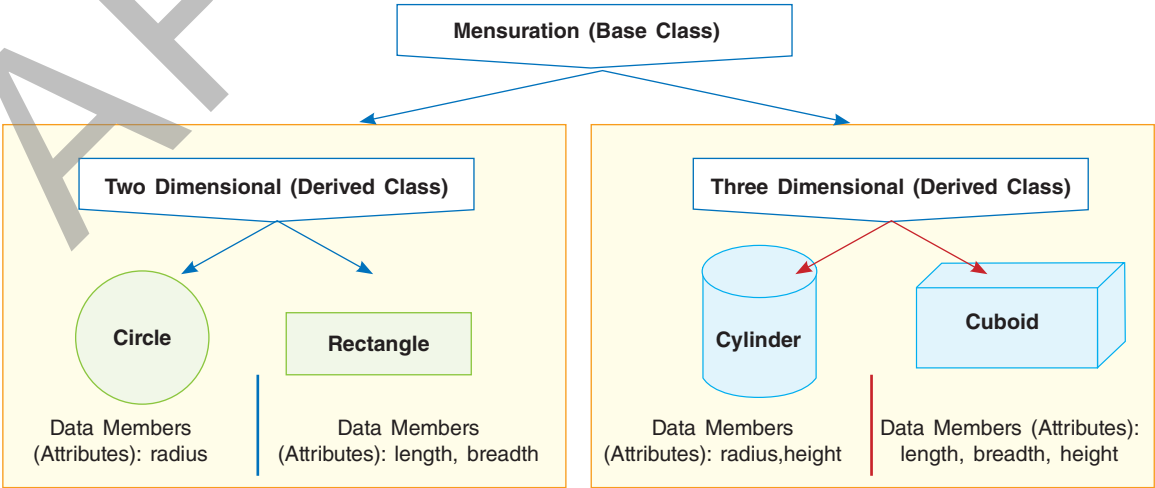
The function provides an interface between data items of the objects and the calling program. Thus, we can say that the function used in a class can only access the data items. The insulation of data that does not allow it to be accessed directly outside the class premises, although it is available in the same program, is known as ‘Data Hiding’.

Hence, wrapping of data and functions of an object as a unit that can be used together in a specific operation is known as ‘Encapsulation’.

• **Inheritance**

While programming, we may need to create multiple classes to meet our requirements. Under such circumstances, it may happen that some of the features of one class are shared by another class. The mechanism in which one class acquires the features of another class is called inheritance. The class that is inherited is known as a Superclass or Base class and one that inherits from a base is called as Subclass, Derived class or Target.

Class is a blue print or prototype that describes the data members and member functions to be used in various operations. The objects of a class possess the data and functions described within the class.



In the given illustration, ‘Mensuration’ is the Base class whose features (properties) are shared by the derived classes ‘Two Dimensional’ and ‘Three Dimensional’ to calculate parameters of different shapes.

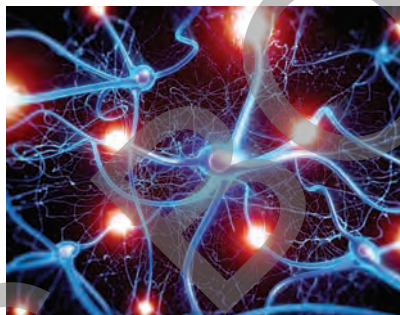
With the help of inheritance, the components in the *Base* class can be reused to perform some other task in *Derived* class. This phenomenon is termed as ‘*Reusability*’. The reusability feature possesses the following advantages in programming:

- Allows less time in writing a program.
- Takes less memory space for storage.
- Enables program execution faster.

Hence, inheritance is a property by virtue of which one class acquires some features from another class. It promotes reusability.

### • Polymorphism

Consider a term ‘cell’. The term ‘cell’ in Physics refers to a device that supplies current in the electrical circuits. In Biology, it means a fundamental and functional part of human body/plant. The same term is also referred to a small room such as Administrative Cell, Examination Cell, etc.



In object oriented programming, an object may include a function that can be used for multiple operations. Using a single function to carry multiple tasks is said to be ‘Polymorphism’.



**Q.1** Name two categories of programming concepts.

**Ans.** The two categories of programming concepts are:

- (i) Procedure Oriented programming
- (ii) Object Oriented programming

**Q.2** Name two procedure oriented and two object oriented programming languages.

**Ans.** (i) Two procedure oriented languages are FORTRAN and BASIC.

(ii) Two object oriented languages are Java and Python.

**Q.3** Name four basic principles of object oriented programming.

**Ans.** (i) Data Abstraction

(ii) Polymorphism

(iii) Inheritance

(iv) Encapsulation

**Q.4** Name an object oriented principle that allows a function to be used for multiple purposes.

**Ans.** Polymorphism

**Q.5** What is meant by 'Base' class and 'Derived' class?

**Ans.** The class from which other classes are derived through the process of inheritance is called base class or super class. The class that inherits properties from a base class is called a derived class.

**Q.6** What does reusability mean?

**Ans.** During inheritance, the components used to perform a task in base class are in turn used in the derived class for other type of processing. This feature is known as reusability.

**Q.7** State the Java concept that is implemented through:

(i) A superclass and a subclass

(ii) The act of representing essential features of a class without including the background details. [ICSE 2013]

**Ans.** (i) Inheritance

(ii) Data Abstraction

**Q.8** Define Encapsulation.

[ICSE 2016]

**Ans.** The wrapping of data and function of a class so that they can be used as a unit is termed as Encapsulation.

**Q.9** What is Inheritance?

[ICSE 2017]

**Ans.** Inheritance is an OOP principle according to which a class acquires some features from another class. It promotes a characteristic called reusability.

**Q.10** What is data abstraction? Explain with an example.

[ICSE 2018]

**Ans.** It is an act of representing essential features of a class without undergoing background details.

For example, while driving a car, you are only aware of its important parts (using essential features) viz. clutch, brake, gears and accelerator. By applying brakes, you can stop the car and thus you need not require to know the internal mechanism of the brake of car.





## EXERCISES

### I. Tick the correct option:

1. In object oriented programming, the stress is given on:  
(a) procedure (b) methods (c) class (d) data
2. Which of the following principle does not allow to access directly from outside the class premises:  
(a) data hiding (b) encapsulation (c) abstraction (d) all of the above
3. The process of combining data and functions that enables them to be together as a single entity is called:  
(a) inheritance (b) encapsulation (c) classification (d) attributes
4. The process by which a class acquires the properties from another class is called:  
(a) polymorphism (b) inheritance (c) abstraction (d) object
5. In procedural programming, the stress is laid on:  
(a) class (b) function (c) data (d) object
6. A ..... is a set of objects that has common attributes and common behavior.  
(a) abstraction (b) encapsulation (c) class (d) function
7. Which of the following is not an object oriented programming language?  
(a) C++ (b) Simula (c) BASIC (d) Java
8. The process of restricting the free flow of data from the outside world is known as:  
(a) encapsulation (b) inheritance (c) function (d) class

### II. Fill in the blanks:

1. .... is the fundamental concept in object oriented programming language.
2. Objects can communicate with each other through ..... in object oriented programming language.
3. Using a function for multiple operations is called as .....
4. .... is an act of representing essential features without including background details.
5. Feature of wrapping ..... and ..... as a single unit is called encapsulation.
6. .... principle of object oriented programming promotes reusability feature.

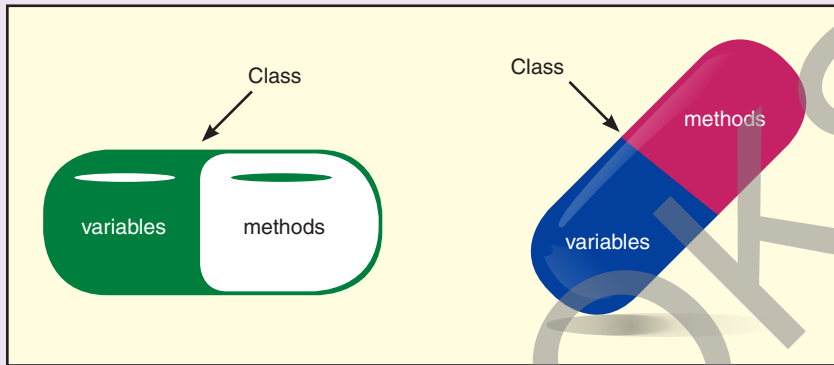
### III. Write short answers:

1. What is object oriented programming? Name two object oriented programming languages.
2. Name four basic principles of object oriented programming.
3. Why do we prefer object oriented approach in complex programming? Explain.
4. What is meant by a base class and a derived class?
5. Mention two limitations of procedure oriented programming approach.
6. What is meant by encapsulation?

[ICSE 2006]



7. Define the following with an example each.
- (a) Inheritance
  - (b) Polymorphism
- [ICSE 2008]
8. In what way is Data Hiding related to Data Abstraction?
9. Give an example to explain Data Abstraction.
- [ICSE 2018]
10. What is meant by Data Hiding?
11. Which of the Object Oriented programming principles explain the following illustrations? Justify.
- (a) The variables and methods are put together in a Class.



- (b) A man withdrawing money from ATM





## Unit II: Elementary Concept of Objects and Classes

### LEARNING SCOPE

- Introduction
- Real world objects and software objects
- Characteristics and behaviour of real world objects and software objects
- Class is an object factory
- Object is an instance of a class
- Class is a user defined data type

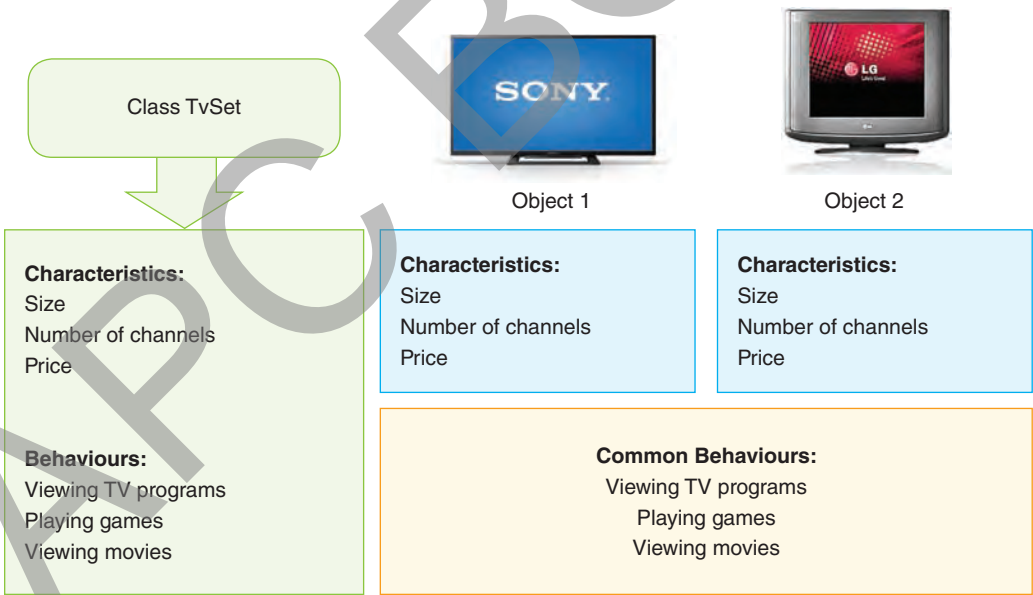
### Introduction

In real world, the class is a set or category of things sharing common characteristics or qualities. Each item belonging to a specific category possesses the characteristics and behaviours described within it.

If we consider ‘furniture’ as a category then different types of furnitures such as table, chair, bench, etc. can be referred to as items of category ‘furniture’. In software terms, category and items are considered to be class and objects respectively.

Hence, class is a blue print/template/prototype that describes the characteristics and behaviours for the similar types of objects. Objects are the entities which possess different characteristics and common behaviour described within the class.

Let us consider an example of real world class to understand the way different characteristics and common behaviours are embedded in its objects.



In the given illustration, some characteristics and behaviours of TV sets are described under class named as ‘TvSet’. The objects of class ‘TvSet’ can be created as shown below:

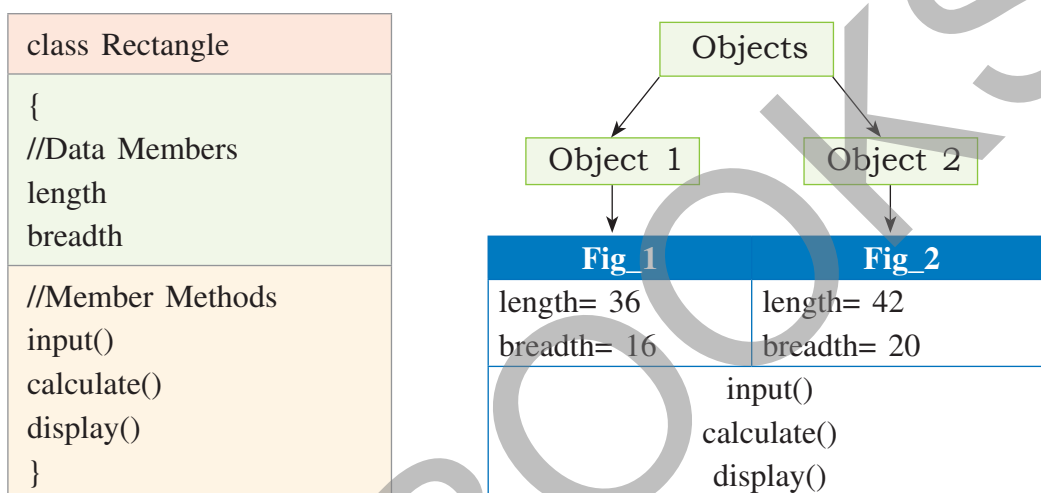
```
TvSet SONY = new TvSet();  
TvSet LG = new TvSet();
```



The moment the objects are created, they will implicitly possess the characteristics and behaviours described within the class. Here, it is to be noted that the objects SONY and LG will contain different characteristics (Size, Number of channels and Price) but they will possess common behaviours such as ‘Viewing TV programs’, ‘Playing games’ and ‘Viewing movies’. In this way, we can say that the class and objects are inter-related to each other.

When class and objects are used in programming, they are known as software class and objects. The description of class and declaration of the objects will take place in the same way as the real world class and objects take (already discussed above).

Let us take an example to understand as how software class and objects will be used for programming purpose.



In the structure shown above, the class 'Rectangle' is designed with the data members (length and breadth) and member functions (input(), calculate() and display()). The objects of the class can be created as shown below:

```
Rectangle Fig_1 = new Rectangle();
```

*Rectangle Fig 2 = new Rectangle();*

The structure shown above, states when the objects Fig\_1 and Fig\_2 are created then each object will automatically possess the data members and member functions described within the class 'Rectangle'. The allocation of data members pertaining to each object can be different but the functions to be carried on them will be common.

Hence,

`Fig_1.input()` : will accept length and breadth of object Fig\_1.

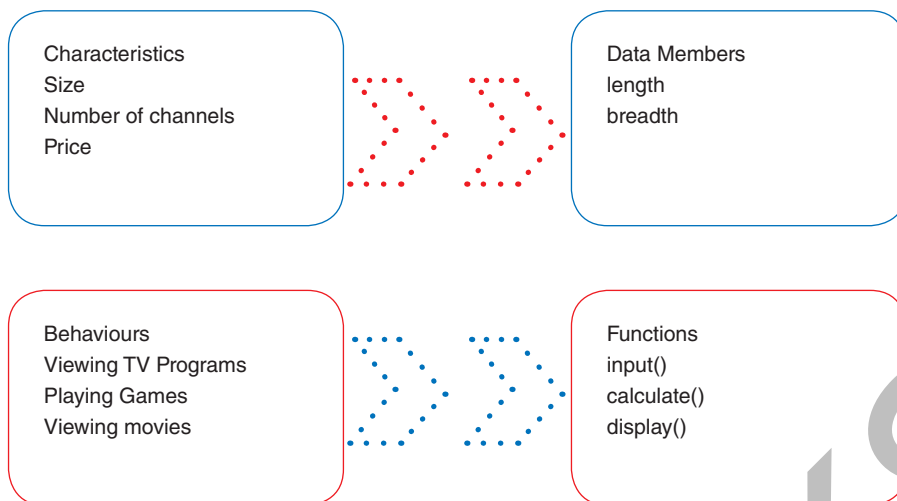
Fig\_1.calculate() : will perform calculations by using length and breadth of object Fig\_1.

Fig\_1.display() : will display the result of calculation of object Fig\_1.

Similarly, you can show the above tasks for the object Fig\_2.

## Real World vs Software Class and Objects

From the real world and software examples of class and objects shown above, it is evident that the characteristics and behaviours of real world class and objects are treated as data members and member methods of software objects respectively.



### Some Meaningful Terms

- Class is an object factory – A class is used to create similar objects that possess different characteristics and common behaviours. Hence, class is called an object factory.
- Object is instance of a class – The data members declared within a class are also known as instance variables. When an object of a class is created, it includes instance variable described within the class. This is the reason that object is called as instance of a class.
- Class is a user defined data type – When user creates a class it becomes a data type for his program. Thus, class is referred to as a user defined data type. This data type includes different primitive types such as int, float, char, etc. Hence, it is also said to Composite Data Type.

#### Points to Remember!

A class creates many objects and interacts by invoking methods.

An object always defines the following:

1. State : It is represented by attributes of an object. It also reflects the properties of an object.
2. Behaviour : It is represented by methods of an object. It also reflects the relation of an object with other objects.
3. Identity : It gives a unique name to an object and enables one object to interact with other objects.

A class is described with some characteristics and behaviours where characteristics are termed as data members (Attributes) and behaviours as the member functions (member methods). Each object belonging to a class contains the attributes (states) and functions. It means that the attribute deals with the contents of an object to make the object more distinct. Attributes are the characteristics or data members of an object.



**Q.1** Name the keyword that defines the characteristics or the data members for the objects of a class.

**Ans.** Attribute

**Q.2** Class and Object are inter-related. Justify with reference to object oriented programming.

**Ans.** A ‘Class’ is used to create various objects that possess different characteristics and common behaviours. Each object follows all the features defined within the class. That is why class is also referred to as a blue print or prototype of an object. This way we can say that they are inter-related.

**Q.3** What is the importance of the keyword ‘new’ while creating an object?

**Ans.** ‘New’ is used for dynamic allocation of the object, i.e., it allots space in the dynamic memory for storage of an object.

**Q.4** In what way would you relate a real world object to a software object?

**Ans.** A real world object contains characteristics and behaviour. The software object replaces characteristics with data members, and replaces behaviour with functions or methods. In this way real world objects and software objects are interrelated.

**Q.5** Why is a class called an object factory? [ICSE 2009]

**Ans.** Class is the prototype of an object. Each object belonging to a specific class possesses the data and functions defined within the class. It also produces the objects of similar type. Hence, a class is termed as an object factory.

**Q.6** Why is a class known as composite data type? [ICSE 2009]

**Ans.** A class is defined with a number of data members of different types. When an object is created, it contains all the members described within the class. The class becomes such a data type that includes various primitive types. Hence, the class is said to be a composite data type.

**Q.7** Why is an object called an instance of a class? [ICSE-2010]

**Ans.** Since, an object possesses instance variables and member methods defined within the class. Hence, an object is called an instance of a class.

**Q.8** What is the difference between an object and a class? [ICSE-2011]

**Ans.** The differences between an object and class are:

Class	Object
1. It is a a blue print or prototype of an object.	1. It is a unique entity having some characteristics and behaviours.
2. It is known as ‘Object Factory’.	2. It is known as an ‘Instance of a Class’.

**Q.9** Write a Java statement to create an object mp4 of class digital. [ICSE 2013]

**Ans.** Java statement to create an object mp4 of class digital:  
digital mp4 = new digital( );



## EXERCISES

### I. Fill in the blanks:

1. Creating ..... is the fundamental concept in object oriented programming language.
2. A class is also considered as an ..... factory.
3. A real world object deals with characteristics and .....
4. The ..... of a class differs on various characteristics.
5. The characteristics of the ..... objects are considered to be the data members of the ..... objects.
6. An object of a class is created by using a keyword .....
7. Class is a ..... of the objects.
8. ....keyword is used for dynamic allocation of an object.
9. The living things are considered as ..... objects.
10. The different ..... of a class have common behaviours.

### II. Answer the following questions:

1. How will you define a software object?
2. Define class and object with an example.
3. What does the following statement mean? [ICSE-2008]  
Employee staff = new Employee ( );
4. A class is also referred to as 'Object Factory'. Comment. [ICSE-2009]
5. Why is a class known as composite data type?
6. A statement is given as:  
'Study\_Table' is an object of the class 'Furniture'. Write down Java statement for the same.
7. Class and Objects are inter-related. Explain.
8. Why is an Object called an 'Instance' of a class? Explain. [ICSE-2010]
9. Write a statement to create an object 'Keyboard' of the class 'Computer'.
10. Mention three characteristics and two methods for the following Classes:  
(a) Class Mobile Phone (b) Class Bike  
(c) Class Fruits (d) Class Pen
11. Design a class program to calculate the discount given to a customer on purchasing LED Television. The program also displays the amount paid by the customer after discount. The details are given as:  
Class name : Television  
Data members: int cost, int discount, int amount  
Member functions:  
Accept( ) : to input the cost of Television  
Calculate( ) : to calculate the discount  
Display( ) : to show the discount and the amount paid after discount



## Unit III: Values and Data Types

### LEARNING SCOPE

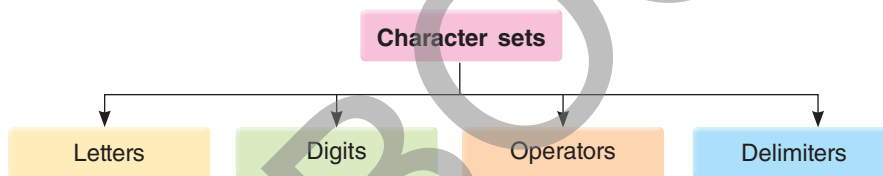
○ Character sets in Java ○ ASCII code ○ Escape sequences ○ Tokens ○ Different types of tokens (Literals, Identifiers, Punctuators, Operators, Keywords) ○ Assignments ○ Constants and variables ○ Initializing a variable ○ Data types in Java: Primitive and Non-Primitive types ○ Type conversion: Implicit and Explicit type conversions.

### Introduction

Program is a set of statements. Each statement of computer program is formed by using different words called 'Token'. Each token is formed by using valid characters of computer language in which program is written. Hence, characters are the basic or fundamental unit of a programming language.

### Character sets in Java

Java being a programming language contains various character sets that can be used while writing a computer program. The character sets used in Java are as follows:



- Letters: All letters are used in Java programming.
  - Upper case – A to Z
  - Lower case – a to z
- Digits: All digits are used in Java programming (i.e., 0 to 9).
- Operators: These are special signs or symbols used to perform operations in Java programming.
  - Arithmetical operators: +, -, \*, /, %, etc.
  - Logical operators: &&, ||, !
  - Relational operators: <, <=, >, >=, ==, !=
- Delimiters: These are special characters used in Java programming.
  - Braces: (, ), {, }, [, ], etc.
  - Punctuation marks: comma(,), dot(.), semicolon(;), ?, :, etc.

### Encoding of characters

You must keep in your mind that the characters entered from the keyboard are not directly stored in the computer memory. It is because the memory stores a character in numeric form (binary number). Hence, the characters entered from the keyboard are encoded into numeric form before actually getting stored in the memory. Basically, two types of encoding schemes are used in computer to encode the characters. They are ASCII and Unicode.

## ASCII Codes vs Unicode

ASCII stands for American Standard Code for Information Interchange. It uses 7 bits code for representing each character. It includes 256 characters but only 128 characters are used while designing and executing a program. ASCII code represents these characters as numbers, with each letter assigned a number from 0 to 127. The ASCII codes of some of the characters used in Java programming are shown below:

Codes	Characters
48 - 57	0 - 9
65 - 90	A - Z
97 - 122	a - z
32	Blank

## Unicode

The Unicode is a wide representation of characters in numeric form. The code contains hexadecimal digits ranging from 0x0000 up to 0xFFFF (i.e., 16-bits code). It can address 1,37,000 characters from 146 modern and historic scripts. It means a code of each character is available under Unicode character set. It is a standard encoding system used to encode a character in any computer language.

## Escape Sequences

There are some characters, known as escape sequences, that are used to control the cursor while displaying the values on the screen. Each escape sequence starts with backslash (\) followed by a specific character. The complete set (backslash + character) is enclosed within double quotes, e.g., “\t”. Normally, a set of characters under double quotes is considered to be a string but backslash (\) acts as a command to control the printing in specific manner. Some of the escape sequences are listed below for the reference:

Escape Sequences	Non-Graphic Character
\t	Horizontal tab
\v	Vertical tab
\\	Backslash
\'	Single quote
\"	Double quote
\b	Backspace
\f	Form feed
\0	Null
\r	Carriage return
\n	New line feed





## Using Escape Sequences

The escape sequences help to control the cursor’s movement on the output screen. It enables the user to get a formatted output.

Some escape sequences which are commonly used in Java programming are explained below:

- (i) “\n” (Backslash n): This character is used for new line feed. As soon as the control encounters “\n”, the cursor skips the current line and moves to the next line on the screen for printing the remaining part of the output.

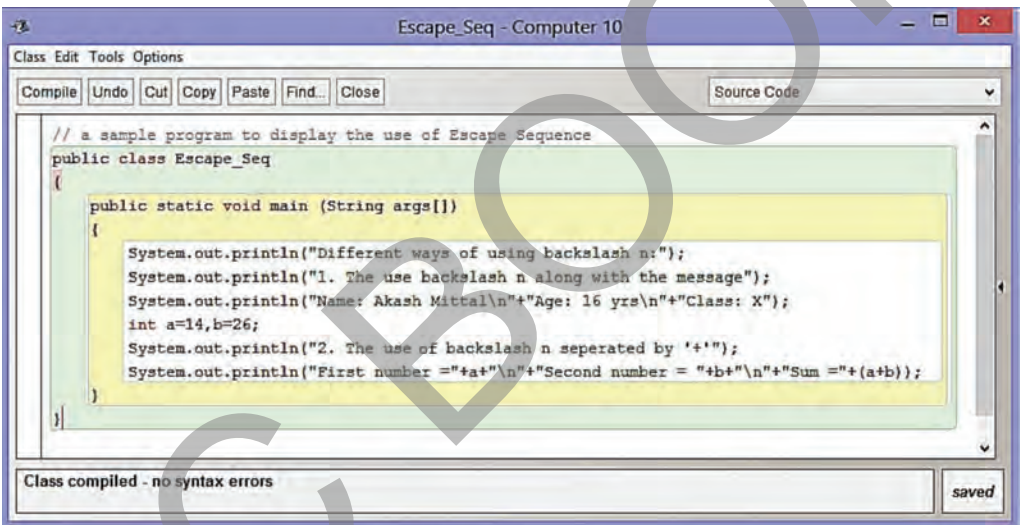
e.g.,

```
System.out.println("Name: Akash Mittal\n"+"Age: 16 yrs\n"+"Class: X");
```

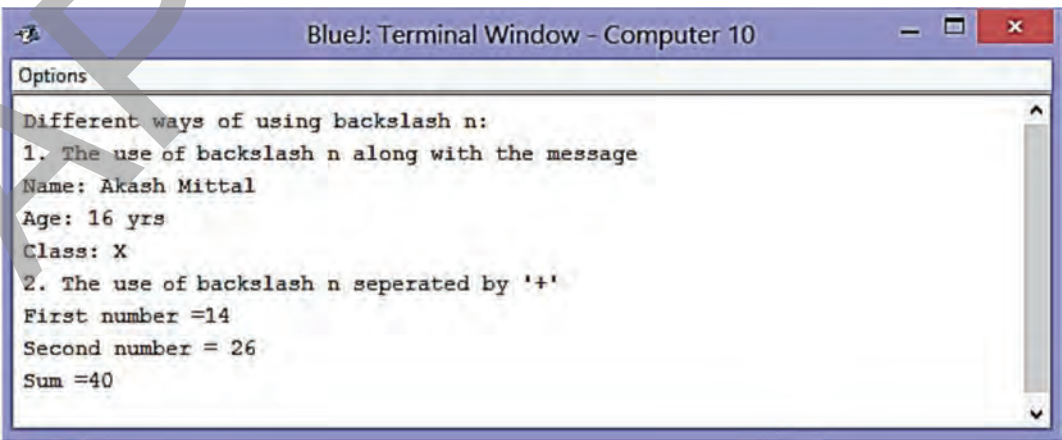
In the print statement shown above, “\n” is included along with the message in double quotes. If the print statement contains variables then you need to use “\n” separately to display the message in different lines, as shown:

```
int a=14,b=26;
```

```
System.out.println("First number =" +a+"\n"+"Second number =" +b+"\n"+"Sum =" + (a+b));
```



The output of the above program is shown below:



- (ii) “\t” (Backslash t): This character is used to separate the values on the screen with a gap of 8 spaces between them. It is also used for printing the result in a formatted pattern.





e.g.,

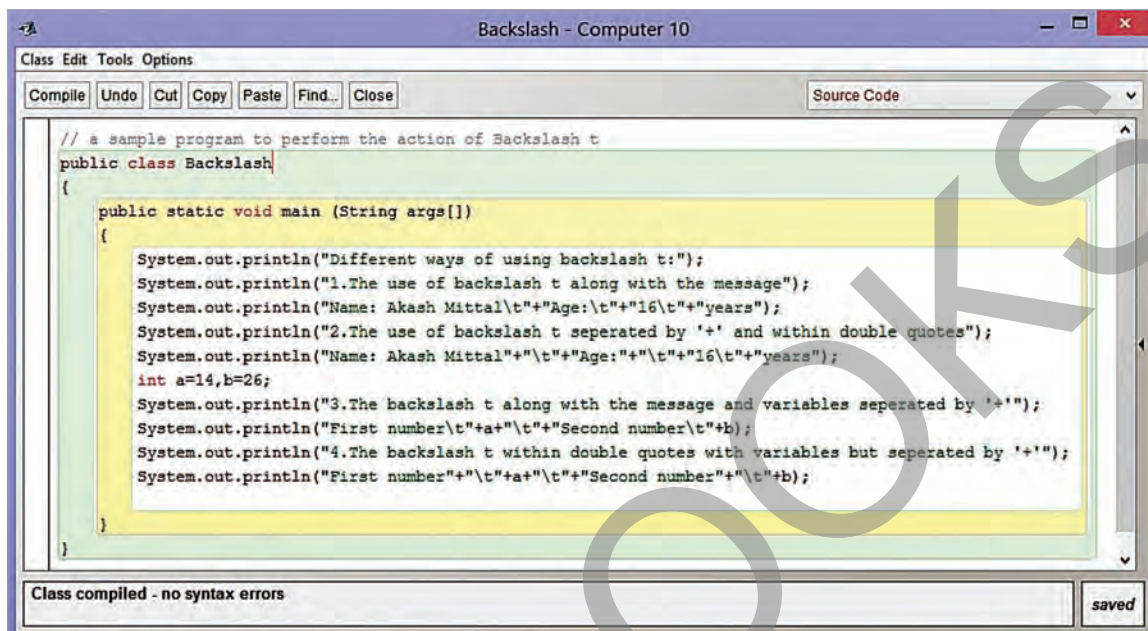
```
int a=14,b=26;
```

```
System.out.println("First number\t"+a+"\t"+"Second number\t"+b);
```

or

```
System.out.println("First number"+"\\t"+a+"\\t"+"Second number"+"\\t"+b);
```

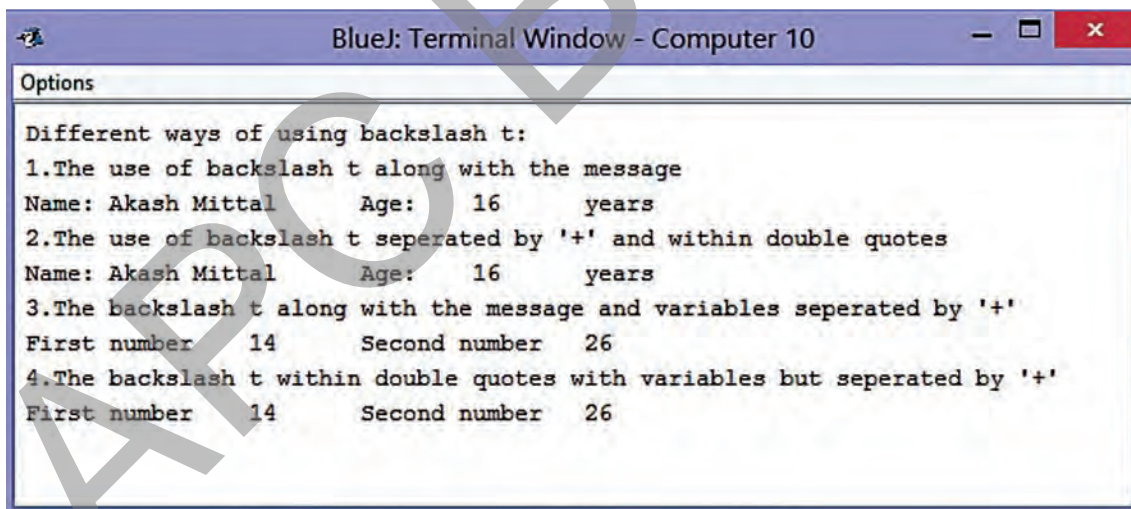
Here, you will notice that the output in both the cases is same.



```
// a sample program to perform the action of Backslash t
public class Backslash {
    {
        public static void main (String args[])
        {
            System.out.println("Different ways of using backslash t:");
            System.out.println("1.The use of backslash t along with the message");
            System.out.println("Name: Akash Mittal\t" + "Age:\t" + "16\t" + "years");
            System.out.println("2.The use of backslash t seperated by '+' and within double quotes");
            System.out.println("Name: Akash Mittal"+"\\t"+"Age:"+ "\\t"+"16\\t"+"years");
            int a=14,b=26;
            System.out.println("3.The backslash t along with the message and variables seperated by '+'");
            System.out.println("First number\t"+a+"\t"+"Second number\t"+b);
            System.out.println("4.The backslash t within double quotes with variables but seperated by '+'");
            System.out.println("First number"+"\\t"+a+"\\t"+"Second number"+"\\t"+b);
        }
    }
}
```

Class compiled - no syntax errors

Thus, the output of the above statements is shown below:



```
Options
Different ways of using backslash t:
1.The use of backslash t along with the message
Name: Akash Mittal      Age:   16      years
2.The use of backslash t seperated by '+' and within double quotes
Name: Akash Mittal      Age:   16      years
3.The backslash t along with the message and variables seperated by '+'
First number   14      Second number   26
4.The backslash t within double quotes with variables but seperated by '+'
First number   14      Second number   26
```

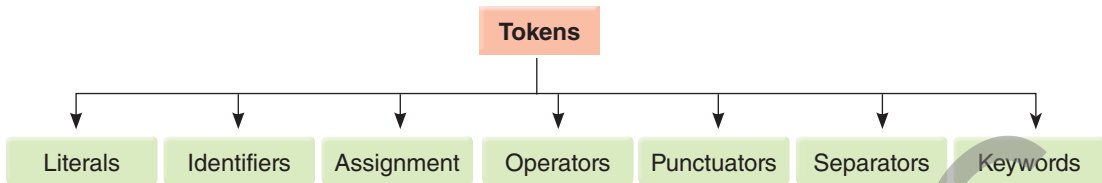
## Tokens

Token is a set of valid characters used for writing a statement in Java program. In other words we may say that a Java statement comprises a number of tokens. They are the smallest elements of the program identified by the compiler.



Thus, token is defined as each individual component of Java program that carries some meaning and takes active part in program execution.

The different types of tokens used in Java are:

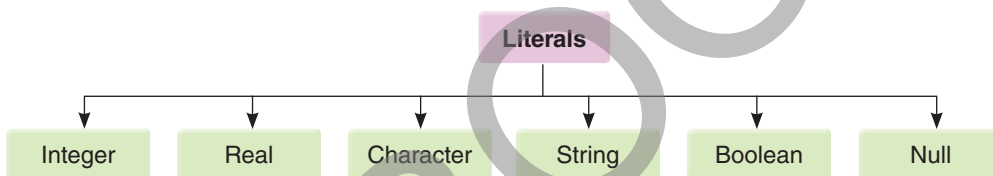


You have already studied about various tokens used in Java programming. Let us have a brief discussion about different tokens.

### • Literals

You would have heard about constants in mathematics. They have fixed values which can't be changed at all. The term constants are referred to as literals in Java programs that remain unchanged during entire execution of the program.

The various types of literals used in Java are as follows:



- **Integer Literals:** The whole numbers (positive or negative) are known as integer literals. They are also called as a fixed point numbers because the decimal point always have a fixed position (i.e., after the last digit of the number). Hence, using decimal point in integer numbers is optional.  
e.g., 16, 345, 8262, etc.
- **Real Literals:** Real literals are also called as floating point numbers. They are the fractional numbers. The placement of decimal point may not be same in real numbers.  
e.g., 3.42, 43.678, 0.000012, 678.4, etc.
- **Character Literals:** A single letter or a digit or any special symbol enclosed within single quotes is known as a character literal. It does not take part in arithmetical calculations.  
e.g., 'B', 'u', '6', '?', etc.
- **String Literals:** A set or group of characters enclosed within double quotes is known as a String literal.  
e.g., "PROCESS", "System", "Salary – 2018", "4357", etc.  
*Note: A string literal can be a set of any type of characters within a limit of 256 characters.*
- **Boolean Literals:** Boolean literals are *true* and *false*. A boolean literal can either be true or false at a time.
- **Null Literal:** Null literal denotes the absence of a value. It is used to initialize an object or array.

e.g.,

```
Integer n = null;
```

```
String s = null;
```

### • Identifiers

Identifiers is a term used for naming a block of statements by which they are identified in a Java program. Identifiers can be a class name, a function name, an object or an interface. Variables are also identifiers because they are used to name a data that holds a value in the memory. The value can change depending upon our requirements in the program.

### • Assignment

The term assignment refers to storing a value in a variable. It is already discussed earlier that the variable and the value to be stored must have compatible data types (same data types).

For example,

```
int a = 12;
```

```
float f = 8.36;
```

```
double d = 3.535345;
```

```
char c = 'E';
```

```
String s = "System";
```

### • Operators

Operators are the symbols or signs used to specify the operations to be performed in a Java expression or statement. There are three types of operators in Java.

(a) Arithmetical Operators: +, -, /, \*, etc.

(b) Relational Operators: <, >, =, !=, <=, etc.

(c) Logical Operators: &&, ||, !, etc.

You study more about operators in the next unit.

### • Punctuators

Punctuators are the punctuation signs used as special characters in Java. Some of the punctuators are comma(,), semicolumns (;), dot (.), etc.

### • Separators

They are the special characters in Java, which are used to separate the variables or the characters.

e.g., Comma(,), Brackets ( ), Curly brackets { }, Square brackets [ ], etc.

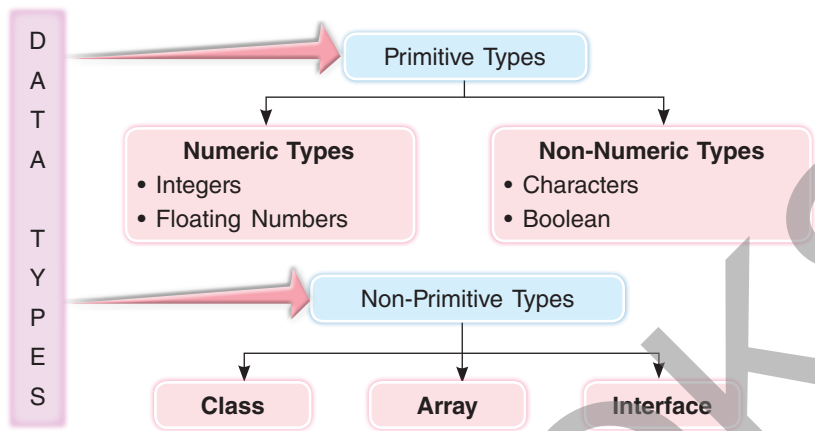
### • Keywords

Keywords are the reserved words which are preserved by the system and carry special meaning for the system compiler. They have already been defined in the language and we cannot use them as names for variables or identifiers, e.g., class, public, throws, for, sqrt, System, etc.



# Data Types

Data type basically refers to as the type of data, a memory location can hold. Whenever, we need to store a value, the respective data type must be mentioned so that the system may structure the memory location for its proper storage. This is necessary for using data types. The classification of data types is as shown below:



## Primitive Data Types

Primitive data types are the basic or fundamental data types used to declare a variable. They are further classified into two categories:

**1. Numeric data types:** The numeric data types are used to declare variables that can store numbers. Based on the types of values that can be stored in a variable, the numeric data type is classified into the following two types:

**Integer type:** A variable declared under this type contains an integer literal. They are further categorised as:

- *byte* : It is used to store integer data up to 8 bits (i.e., 1 byte).  
e.g., `byte b;`  
`b=5;`
- *short* : It is used to store a small range of integers up to 16 bits (i.e., 2 bytes).  
e.g., `short s;`  
`s=25;`
- *int* : It is used for storing integers up to 32 bits (i.e., 4 bytes).  
e.g., `int n;`  
`n = 345;`
- *long* : It is used for storing large integers up to 64 bits (i.e., 8 bytes).  
e.g., `long l;`  
`l = 1234;`

The user may select a particular data type according to the size of the value to be stored.

**Real type:** A variable declared under this type contains floating point data (i.e., the fractional numbers including a decimal point). There are categorised as:

- *float* : It represents a fractional number with a small range up to 32 bits (i.e., 4 bytes).  
e.g., `float f;`  
`f = 6.32;`



- **double** : It represents a fractional number with wide range values up to 64 bits (i.e., 8 bytes).

e.g., double d;  
d = 0.12543679;

The user may decide to declare a variable as float or double as per the size of the values to be stored.

### Precision of the data type

When you divide a number by another number, you may get a non-terminating decimal in some cases. For example, if you divide 5 by 6, it results in 0.83333333..... . Here, digit 3 repeats itself in the decimal part. If you store the result in float data type then the system will not be able to store such an infinite number. A float data type has the capacity to store numbers up to 32 bits (i.e. up to 7 significant digits).

For example: float p = 0.8333333

Double data type has more precision. If you declare p as double data type, it will store up to 15 significant digits.

Hence, p = 0.83333333..... (14 times)

**2. Non-numeric data types:** Non-numeric data types do not deal with numbers rather they are used to declare a character or a set of characters. They can be classified into following two categories:

- **Character type:** A variable is declared to be a character type when it holds a character literal (i.e., a single letter or digit or a special symbol enclosed within single quotes).

e.g., char c;  
c = 'X';

A set of characters enclosed within double quotes is termed as a string. If you want to store a string literal to a variable then the variable must be declared as a String data type.

e.g., String s;  
s = "Computer";

- **Boolean type:** Boolean literals are true or false. To store a Boolean literal, the variable must be declared boolean.

e.g., boolean b;  
b = true;

### Note:

A boolean literal true or false must not be enclosed within double quotes otherwise, it will become a string literal.

### Declaration vs Initialization

Declaration is the term used to declare the type of a variable whereas initialization refers to assigning or storing a specific value to the variable.

e.g., int a; //variable declaration  
a = 5; //variable initialization

First of all, the variable 'a' is declared to hold an integer type data and then it is initialized with the integer value 5.



Default Initialization

If a variable is not initialized but is used in the program, then the system initializes it implicitly with a specific value called Default Initial Value. It is indeed necessary to know what initial value can be stored by default to a variable of specific data type. The table shown below demonstrates data types of different variables and their default values.

Data Type	Declaration	Default Initial Values
integer	int a;	a=0;
float	float f;	f=0.0f;
double	double d;	d=0.0d;
character	char chr;	chr= ‘\u0000’;
string	String st;	st= “”;
boolean	boolean p;	p=false;

The variables which are declared within the class as data members then they are initialized with default initial values. The variable declared under main( ) as local variables need to be initialized by the user.

Here, ‘\u0000’ means blank/space character.

Static and Dynamic Initializations

Static initialization is a way to store a value directly to a variable by using assignment operator (=). You must take care that the data type of the variable must be same as the type of value is to be stored.

e.g.,

Data Type	Declaration	Static Initialization
integer	int a;	a=10;
float	float f;	f=2.45;
double	double d;	d=5.6435789;
character	char chr;	chr= ‘P’;
string	String st;	st= “JAVA”;
boolean	boolean p;	p=false;

Dynamic initialization is used to initialize a variable at run time (i.e., during execution of the statement). In this system, the variable is initialized with the value which is the outcome of some expression or a function.

Data Type	Declaration	Dynamic Initialization
integer	int a,b,c;	c=a+b;
float	float p=2.2,k=4.22,f;	f=p+k;
double	int a=49; double d;	d=Math.sqrt(a);
String	String st1, st2, st3;	st3=st1+st2;



## Non-Primitive Data types

A non-primitive data type is one that is derived from primitive data types. In other words, we may say that a number of primitive data types are used together to represent a non-primitive data type. Hence, a non-primitive data type is also called a composite data type. Some of the examples of non-primitive data types are array, class and interface.

## Type Conversion

Arithmetical expression is defined as a set of variables, constant and arithmetical operators.

e.g.,  $a + b * c - 4$ ;

Here, a, b, and c are the variables, 4 is a constant and +, -, \* are the arithmetical operators.

If an arithmetical expression contains all the data values (variables or constants) of same data types is known as *Pure Arithmetical Expression*.

e.g., `int a,b,c;`  
 $a * b + c/2$

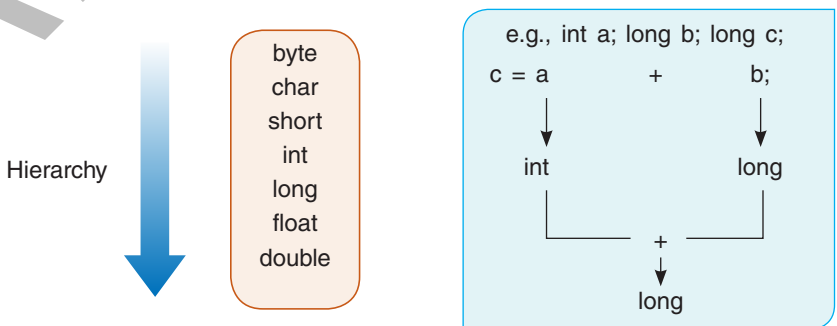
The above expression contains all the values (a, b, c and 2) of same data type, i.e., integer. Hence, it is a pure expression. The expression will obviously result in integer type value.

On the other hand, when the data items in the expression may have different data types then the expression is called as *Impure Arithmetical Expression or Mixed Mode Expression*.

Here, our discussion is based on impure arithmetical expression. If various types of data are used in an expression, a question obviously comes to your mind what type of result it will return after execution. In fact, the system converts the result into a specific data type known as *Type Conversion*. Basically, the data type conversion takes place in two ways as discussed below:

### • Implicit type conversion

In a mixed mode expression, when the data type of the result gets converted into the higher most data type available in the expression without any intervention of the user, is known as implicit type conversion. Implicit type data conversion is also known as Coercion.



• **Explicit type conversion**

It is another type of data conversion in which the data type of the result in a mixed mode expression gets converted into a specific type as per user’s choice. The choice of data type must be written within braces ( ) before the expression. The explicit type data conversion is also known as Type Casting.

```
e.g.,  
int a, float b, char c;  
d = (char) (a + b * c);
```

In the example shown above, a, b and c are different types of data. The result should implicitly be in higher most float type. But, the user demands the result in char type which is a lower type among all the data types. Hence, the system is forced to provide a char type result using type casting.

**QUICK  
Recap**

Assign the following literals of Column A with the variables of Column B.

Column A	Column B
a. “Java Language”	i. int x
b. 345	ii. char p
c. false	iii. String st
d. 45.65	iv. boolean m
e. ‘K’	v. double n
f. 0.004125	vi. float y

**Hint:**

- (a) string
- (b) integer value
- (c) boolean value
- (d) floating point number
- (e) character value
- (f) double floating point number





**Q.1** What is meant by boolean variable?

**Ans.** Boolean variable is a variable that holds boolean value, i.e., true or false.

**Q.2** What is the difference between “true” and true?

**Ans.** “true” (enclosed under double quotes) is a string literal whereas true (without double quotes) is a boolean literal.

**Q.3** State the escape sequences for horizontal tab and new line feed.

**Ans.** Escape sequences for horizontal tab is “\t” and for new line feed is “\n”.

**Q.4** Is String type a primitive or non-primitive data type?

**Ans.** String is a specific data type that is framed using various character data types. It is neither a primitive nor a non-primitive data type.

**Q.5** In what way float initialization is different from double type initialization?

**Ans.** A float type variable is initialized with a fractional value as 0.0L whereas, a double type variable is initialized with a fractional value as 0.0D. The difference between initializations is the suffix, i.e., L and D respectively.

e.g.,

float f=0.0F;

double d=0.0D;

**Q.6** What is a variable?

**Ans.** Variable is a named memory location that is used to contain a value that can change depending on the circumstances during the execution of a program.

**Q.7** Differentiate between pure and impure type expressions.

**Ans.** A pure expression contains the elements of the same data type whereas, an impure expression contains elements of different data types.

**Q.8** What is meant by type conversion? How is an implicit conversion different from an explicit conversion? [ICSE 2010]

**Ans.** In a mixed mode expression, the result needs to be obtained in a single specific data type. Hence, the data type is converted to a required type. This process of conversion is known as type conversion.

In an implicit type conversion, the result of a mixed mode expression is obtained in the higher most data type of the variables without any intervention of the user.

e.g., int a; float b; float c;

c = a + b;

In case of explicit type conversion, the data gets converted to a type as per the user's choice and requirement.

e.g., int a; float b; double c;

b = (float)(a \* c);

**Q.9** Give one example each of a primitive data type and a composite data type. [ICSE 2012]

**Ans.** Examples of primitive data type: int, float, double, long, char, etc. (*any one*)

Examples of composite data type: array, class, interface, etc. (*any one*)

**Q.10** State the values of n and ch.

char c='A';

int n=c+1;

char ch=(char)n;

[ICSE 2012]

**Ans.** The value of

(i) n = 66

(ii) ch = B



- Q.11** What is a literal? [ICSE 2013]
- Ans.** Literal is a constant that remains fixed throughout the discussion of a program. It is the value that can be assigned to a variable. For example,  
 Integer literals: 12, 4, 342, etc.  
 Real literals: 1.02, 345.657, 0.00002, 1.0e-05, etc.  
 Character literals: 'A', 'p', ':', '?', etc.  
 String literals: "COMPUTERS", "PAY2013", "3421", etc.  
 Boolean literals: true or false
- Q.12** What are the types of casting shown in the following examples?  
 double x = 15.2;  
 int a = 12;  
 (i) int y = (int) x;  
 (ii) long b = a; [ICSE 2013]
- Ans.** (i) Explicit type casting  
 (ii) Implicit type casting
- Q.13** Name the primitive data type in Java that is:  
 (i) A 64-bit integer and is used when you need a range of values wider than those provided by int.  
 (ii) A single 16-bit Unicode character whose default value is '\u0000'. [ICSE 2014]
- Ans.** (i) long  
 (ii) char
- Q.14** State one difference between the floating point literal and double type literal. [ICSE 2014]
- Ans.** Floating point literal is a fractional number with short precision and the values can range from -3.4E+38 to 3.4E+38, whereas double type literal is used for long precision and the values can range from -1.7E+308 to 1.7E+308.
- Q.15** What are the default values of the primitive data type int and float? [ICSE 2015]
- Ans.** The default value of int is 0 (zero) and float is 0.0F.
- Q.16** Identify the literals listed below.  
 (i) 0.5 (ii) 'A'  
 (iii) false (iv) "a" [ICSE 2015]
- Ans.** (i) Real (ii) Character  
 (iii) Boolean (iv) String
- Q.17** Arrange the following primitive data types in an ascending order of their size:  
 (i) char (ii) byte  
 (iii) double (iv) int [ICSE 2015]
- Ans.** byte, char, int, double
- Q.18** What are the types of casting shown by the following examples? [ICSE 2016]  
 (i) char c = (char) 120; (ii) int x = 't';
- Ans.** (i) Explicit (ii) Implicit
- Q.19** Write one difference between primitive data types and composite data types. [ICSE 2016]
- Ans.** The fundamental or basic data types (viz. int, float, char) that are provided in the language are known as primitive data types. Whereas, derived data types or non-primitive data types are called as composite data types, e.g., class, array.

## EXERCISES

### I. State whether the following statements are 'True' or 'False':

1. There are 128 set of different characters used in a Java program.
2. The ASCII codes of upper case letters range from 97 to 122.
3. A variable gives the exact representation of data.
4. The data types int, float, char are called non-primitive types.
5. A String literal is assigned to a String variable.
6. A character literal is always enclosed in double quotes.
7. String constant can be written by using a set of alphanumeric characters.
8. An object is said to be a non-primitive data.
9. The data type int stores fractional values.
10. Boolean type data is used to test a condition and results in either true or false.


### II. Write short answers:

1. What is meant by data type? Name two types of data type.
2. Why is it necessary to define data type in Java programming?
3. Define the following with an example:
 

(a) variable	(b) constant
(c) boolean data type	(d) coercion
(e) primitive data type	(f) non-primitive data type
4. What is a token? Name different types of tokens.
5. Explain the term type casting. [ICSE 2007]
6. Assign the following to a variable with suitable data type.
 

(a) $m = \frac{22}{7}$	(b) $p = 1.4142135$ (value of square root of 2)
(c) $k = 0.00004545$	(d) $n = 24.50$
7. Distinguish between:
 

(a) Token and Identifier	[ICSE 2008]
(b) Character and Boolean literal	[ICSE 2009]
8. Explain the term type conversion. How is implicit conversion different from explicit conversion? [ICSE 2010]
9. Classify the following as primitive or non-primitive data types.
 

(a) char	(b) arrays
(c) int	(d) classes

 [ICSE 2018]
10. In what way is static initialization of data type different from dynamic initialization?
11. Predict the return data type of 'r' and 'n' from the snippet:
 

```
(a) int p; (b) float m;
    r = p+m;
    n = m/3*(Math.pow(4,3));
    System.out.println(r);
    System.out.println(n);
```
12. Give reason whether the following assignments are correct or not:
 

(a) <code>int m = 155;</code>	(b) <code>float f = 0.002654132;</code>
(c) <code>String str = 'Computer';</code>	(d) <code>boolean p = false;</code>
(e) <code>String b = "true";</code>	(f) <code>char ch = "apps";</code>
(g) <code>String st= "Application";</code>	(h) <code>double n= 455.29044125;</code>



## Unit IV: Operators in Java

### LEARNING SCOPE

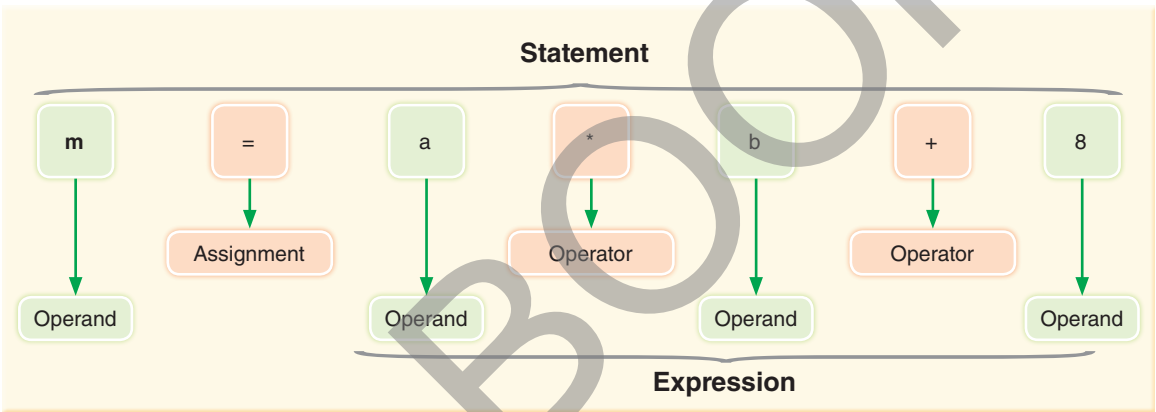
- Expression and Statement
- Types of Operators: Unary, Binary and Ternary; Arithmetical, Logical and Relational; Increment and Decrement; and Postfix and Prefix operators

You have already learnt that an **operator** is a symbol or sign used to specify an operation to be performed in Java programming. It represents the action to be taken whereas, the elements used in the operation on which the action is to be taken are known as **operands**.

### Expression and Statement

Expression is a set of variables, constants and arithmetical operators. In other words, an expression is a combination of operators and operands.

Let us consider an expression as  $a * b + 8$ .



When an expression is assigned to a variable, the complete set is referred to as an arithmetical statement. Hence,  $m = a * b + 8$  is an arithmetical statement.

### Mathematical expression vs Java expression

Some of the operators used in Java programming are different than the mathematical operators. Hence, writing an expression in Java will differ from mathematical expression.

Here, few examples are given below to illustrate how mathematical expressions are written in Java.

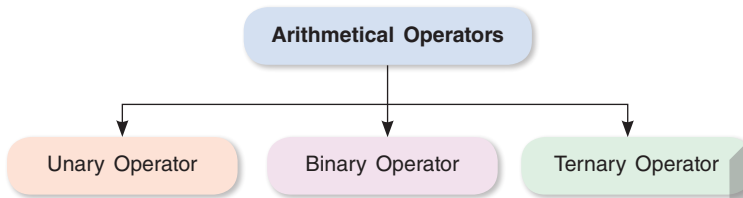
Mathematical Expressions	Java Expressions
$\frac{22}{7}r^2$	<code>(22/7)*r*r</code>
$\frac{(a+b)^2}{a-b}$	<code>(a+b)*(a+b)/(a-b)</code>
$\frac{2}{3}a^2 + \frac{0.05}{b} + c^3$	<code>2/3*a*a+0.05/b+c*c*c</code>
$\frac{y}{6} + \frac{2y}{3} - \frac{y^2}{5}$	<code>y/6+(2*y)/3- (y*y)/5</code>
$\frac{2}{ab} + \frac{3}{c^2} + \frac{5}{abc}$	<code>2/(a*b)+3/(c*c)+5/(a*b*c)</code>

## Types of Operators

Java uses three types of operators. They are arithmetical, logical and relational operators. Let us discuss them in brief.

### Arithmetical Operators

The operators that are used to perform arithmetical operations on the operands are known as arithmetical operators. The arithmetical operators are further categorised into three types:



#### Unary Operator

An arithmetical operator that operates on a single operand is known as Unary Operator. The different types of unary operators are:

##### Unary plus (+)

This operator is used before the variable to keep its sign unchanged. Writing + (unary operator) before an operand is optional.

e.g., if  $a = 4$  then  $+a$  will result in 4.

If  $a = -9$  then  $+a$  will result in -9.

##### Unary minus (-)

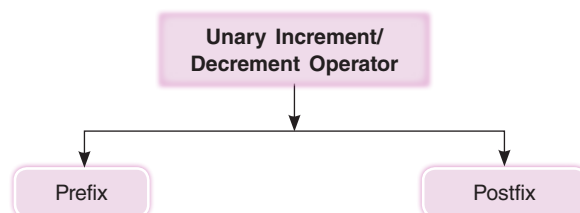
This operator is also used before the operand. It reverses the sign of the operand before which it is used.

e.g., if  $a = 6$  then  $-a$  will result in -6.

If  $a = -3$  then  $-a$  will result in 3.

##### Unary Increment (++) and Decrement (--)

The unary increment and decrement operators used with single operand will increase and decrease the value of an operand by 1 respectively. Based on the position of its use the unary increment and decrement operators are further classified into prefix and postfix operators.



- **Prefix**

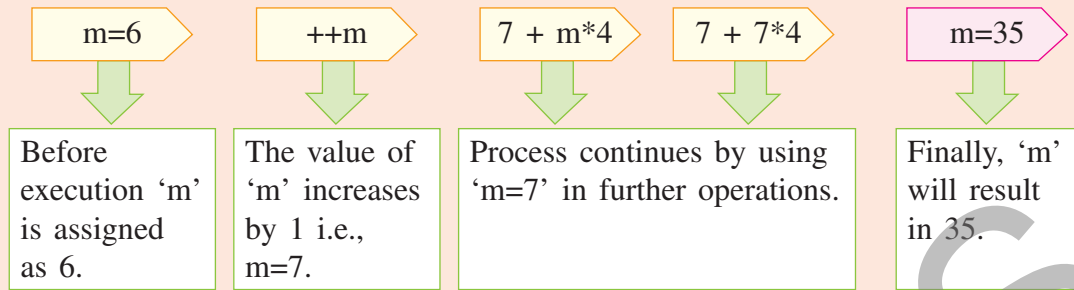
When the increment or decrement operators are written before the operand, they are termed as prefix operators. They increase or decrease the value before the action has taken place. Let us see how they work.

- Prefix increment: In this case, the increment operators (++) are written before the operand.



### Execution of Prefix increment:

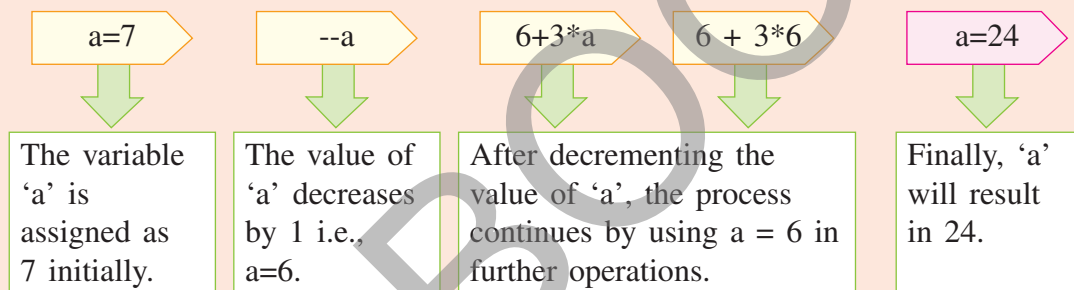
```
int m = 6;  
m = ++m + m* 4;
```



- Prefix decrement: In this case, the decrement operators ( $--$ ) are written before the operand.

### Execution of Prefix decrement:

```
int a = 7;  
a = (--a)+3*a;
```



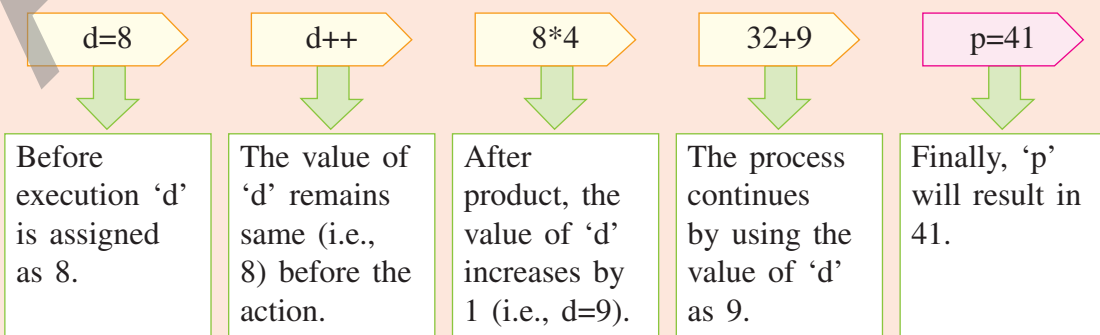
### • Postfix

When the increment or decrement operators are written after the operand, they are termed as postfix operators. They increase or decrease the value after the action has taken place. Let us see how they work.

- Postfix increment: In this case, the increment operators ( $++$ ) are written after the operand.

### Execution of Postfix increment:

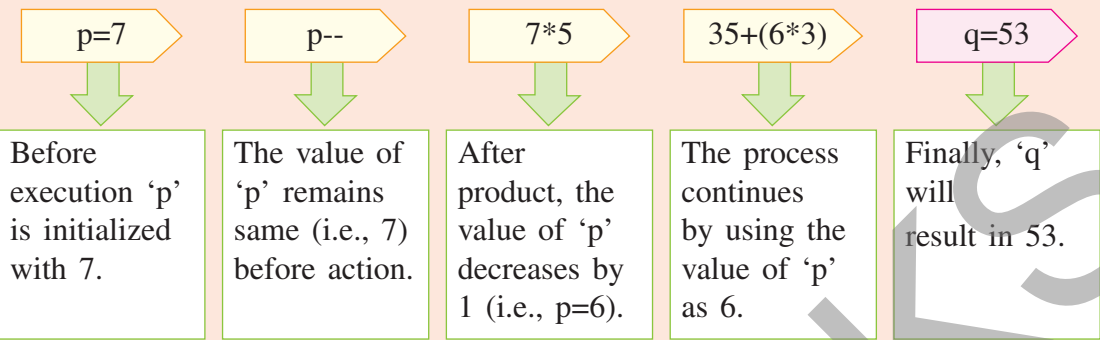
```
int d = 8;  
int p = (d++* 4)+d;
```



- Postfix decrement: In this case, the decrement operators (--) are written after the operand.

**Execution of Postfix decrement:**

```
int p = 7;  
int q = p--*5+ p*3;
```



**Binary Operators**

The operators that operate on two operands are said to be binary operators. The binary operators are +, -, \*, / and %. These operators are used to perform fundamental arithmetic operations such as addition, subtraction, multiplication, division, etc.

A table is given below to illustrate binary arithmetic operators or binary operators:

Operators	Symbols	Format
Addition	+	a+b
Subtraction	-	a-b
Multiplication	*	a*b
Division	/	a/b
Modulus/Remainder	%	a%b

**Shorthand operations**

The binary operators available in Java language can also be written in short form. The table shown below illustrates the application of shorthand operators:

Expression	Shorthand form
q = q/8	q /= 8
m = m % 3	m %= 3
p = p*5	p *= 5
a=a-b	a-=b
m= m+n	m+=n

You already know that an accumulator is a variable that contains cumulative sum of a number of values and counter is a variable that increases by one after each execution.

A shorthand operator can be used only when a variable acts as an accumulator or a counter.



## Ternary Operators (Conditional Operator)

Ternary operators operate on three operands. It is also called conditional operator because the value assigned to a variable depends upon a logical expression or a condition.

### Syntax:

variable = (test condition)? Expression 1: Expression 2.

The variable will be assigned with the value of 'Expression 1' or 'Expression 2' if the test condition is true or false respectively.

For example:

a = 5; b = 3;

max = (a>b)? a:b;

Here, the value 5 is stored in max, as the test condition a>b is true.

min = (b>a)? a:b;

In this case, value 3 is stored in min, as b>a is false.

## SOLVED EXAMPLES

1. Rewrite the snippet by using Ternary operators:

```
if((a+b)>c)
{
    String st= "Third number is smaller";
}
if((a+b)<c)
{
    String st= "Third number is greater";
}
```

**Ans.** String st = (a+b>c)? "Third number is smaller": "Third number is greater";

2. Rewrite the following program segment using if....else statement.

comm = (sale>15000) ? sale\*5/100 : 0;

[ICSE 2013]

**Ans.** Snippet by using if....else statement:

```
if(sale > 15000)
    comm = sale * 5/100;
else
    comm = 0;
```

### • Nested Ternary Operator

When a conditional operator is used within another conditional operator, it is called as 'Nested Conditional Operator'.

For example:

A program snippet to find the greatest number among three given numbers using nested ternary operator:

int a = 5, b = 12, c = 3;





$\text{max} = \underbrace{(a > b) ?}_{\text{Test condition}} \underbrace{(a > c) ? a : c}_{\text{Expression 1}} : \underbrace{(b > c) ? b : c}_{\text{Expression 2}};$

The test condition  $a > b$  is false. Hence, it will operate expression 2. In expression 2, the test condition  $b > c$  is true which enables value 12 to be stored in max.

Hence,  $\text{max} = 12$ .

### • Relational Operators

There are some operators that are used to show the relation among the operands such as less than, greater than, equal to, not equal to, etc. They are basically used to compare the values of two operands and results in True or False accordingly. The different types of relational operators are as follows:

Symbol	Meaning	Syntax
<	Less than	$a < b$
>	Greater than	$a > b$
< =	Less than or equal to	$a < = b$
> =	Greater than or equal to	$a > = b$
= =	Equal to	$a == b$
! =	Not equal to	$a != b$

### • Logical Operators

The logical operators are the operators that are used between two or more conditions in a Java statement. They result in true or false depending upon the values in all the conditions. The logical operators used in Java are && (AND), || (OR) and !(NOT).

#### • Logical AND (&&)

This operator results in true if all the conditional operands associated with it are true.

e.g.,

$(a > b) \&\& (a > c)$

The following table illustrates the use of logical AND (&&) operator.

$a > b$	$a > c$	$(a > b) \&\& (a > c)$
False	False	False
False	True	False
True	False	False
True	True	True

#### • Logical OR (||)

This operator returns false when all the conditional operands associated with it are false, otherwise true.

e.g.,

$(a > b) || (a > c)$



Let us understand its return value with a table shown below:

a > b	a > c	(a > b)    (a > c)
False	False	False
False	True	True
True	False	True
True	True	True

• **Logical NOT (!)**

Logical NOT operator used along with a conditional operand reverses the result. It means, if the condition is false it will result in true and vice-versa.

e.g.

!(a>b)

a > b	!(a > b)
False	True
True	False

**Points to Remember!**

Precedence of logical operators is NOT (!), AND (&&) and OR (!!). When a statement contains all the three logical operators then NOT operator will be performed first.

Example:

if ((a>b)    && (b>c)    || (b>a)    ! (c>b))

↓

2

↓

3

↓

1

Order of the operations is: 1 → 2 → 3

NOT

AND

OR

SOLVED EXAMPLES

**Example 1** If y = 14 then find z = (++y \* (y++ + 5))

**Solution**

z = (++y \* (y++ + 5))

= (15 \* (15 + 5))

= (15 \* (20))

= (15 \* 20)

= 300

**Example 2** If  $a = 8$ ; find the value of  $a- = ++a + a++ + 4$

**Solution**

$$\begin{aligned} a- &= ++a + a++ + 4 \\ a &= 8 - (9 + 9 + 4) \\ &= 8 - 22 \\ &= -14 \end{aligned}$$

**Example 3** If  $a = 12, b=8$ ; find the value of  $a* = ++a/6 + b++ \% 3$

**Solution**

$$\begin{aligned} a* &= ++a/6 + b++ \% 3 \\ a &= 12*(++a/6 + b++ \% 3) \\ &= 12*(13/6 + 8\%3) \\ &= 12*(2+2) \\ &= 48 \end{aligned}$$

**Example 4** If  $x = 4$ ; find the value of  $x+ = x++ * ++x \% 2$

**Solution**

$$\begin{aligned} x+ &= x++ * ++x \% 2 \\ x &= 4 + (4*6) \% 2 \\ &= 4 + (24\%2) \\ &= 4 + 0 \\ &= 4 \end{aligned}$$

**Example 5** If  $a = 48, b =13$ ; find the value of  $a+ = b++ * 5 / a++ + b$

**Solution**

$$\begin{aligned} a+ &= b++ * 5 / a++ + b \\ a &= 48 + (13 * 5)/48 + 14 \\ &= 48 + (65 / 48) + 14 \\ &= 48 + 1 + 14 \\ &= 63 \end{aligned}$$


**Q.1** What are the values of x and y when the following statements are executed?

```
int a=63, b=36;
boolean x=(a>b)?a:b;
int y=(a<b)?a:b;
```

[ICSE 2012]

**Ans.** The given condition statement is incorrect:

```
boolean x=(a>b)?a:b;
It should be :
boolean x=(a>b)?true : false;
then, the values of x and y will be
x = true
y = 36
```

**Q.2** What is meant by the precedence of operators?

[ICSE 2013]

**Ans.** The hierarchical order in which the operators are used for carrying out operation is known as precedence of operator.

e.g., Precedence of arithmetical operators : Brackets, Exponent, Multiplication/ Division, Addition/Subtraction.

Precedence of logical operators : Not, and, or (!, && and ||)

**Q.3** Operators with higher precedence are evaluated before operators with relatively lower precedence. Arrange the operators given below in order of higher precedence to lower precedence.

(i) &&                      (ii) %                      (iii) >=                      (iv) ++                      [ICSE 2014]

**Ans.** The correct order of precedence of operators is:

(i) ++                      (ii) %                      (iii) >=                      (iv) &&

**Q.4** Give the output of the following method.

```
public static void main(String[] args)
{
    int a = 5;
    a++;
    System.out.println(a);
    a - = (a- -) - (- - a);
    a    = 6 - (6 - 4)
        = 6 - 2
        = 4
    System.out.println(a);
}
```

[ICSE 2014]

**Ans.** 6 and +4

**Q.5** Evaluate the value of n, if value of p=5, q=19

[ICSE 2015]

```
int n = (q-p) > (p-q)? (q-p): (p-q);
```

**Ans.** n = 14



## EXERCISES

### I. State whether the following statements are 'True' or 'False':

1. The precedence of operators in Java follows BODMAS.
2. The output of `a++` will be 1, if `int a = -1`.
3. The relational operators always result in terms of 'True' or 'False'.
4. Given: `int m=5; m*=5` then the value stored in `m` results in 55.
5. The statement `(a>b)&&(a>c)` uses a logical operator.
6. If `int a=27,b=4,c=0;` then `c = a%b;` results in 3.
7. The statement `p+=5` means `p=p*5`.
8. In the precedence of logical operators; NOT is followed by AND.


### II. Write the Java expressions for the following:

- |  |             |  |             |
|--|-------------|--|-------------|
| 1. $z = 5x^3 + 2yx + y$                | [ICSE 2011] | 6. $p = a^2 + b^2 + 2ab$                         | [ICSE 2015] |
| 2. $m = \frac{a^2 + b^2}{(a + b)}$     |             | 7. $y = 2(lb + bh + lh)$                         |             |
| 3. $s = ut + \frac{1}{2}at^2$          | [ICSE 2013] | 8. $p = \frac{a}{b^2} + \frac{b}{a^2}$           |             |
| 4. $f = \frac{uv}{(u + v)}$            |             | 9. $z = x^3 + y^3 - \frac{y}{z^3}$               | [ICSE-2006] |
| 5. $d = \frac{\sqrt{3x + x^2}}{a + b}$ | [ICSE 2018] | 10. $q = \frac{1}{\sqrt{a + b}} + \frac{3}{c^2}$ |             |

### III. Predict the output:

1. `int c = (3<4)? 3*4:3+4;`
2. `int a = 14, b=4;`  
`boolean x = (a>b)? true: false;`
3. `int x = 90;`  
`char c = (x<=90)? 'Z':'T';`
4. `int a = 18; int b = 12;`  
`boolean t = (a>20&& b< 15)? true:false;`
5. `c = (val + 550 < 1700)? 200: 400;`  
if: (a) `val = 1000` (b) `val = 1500`

### IV. Answer the following questions:

1. What is an operator? What are the three main types of operators? Name them.
2. How is Java expression different from statement?
3. Explain the following with one example each.
 

(a) Arithmetic operator	(b) Relational operator
(c) Logical operator	(d) Ternary operator
4. Distinguish between:
 

(a) Unary & Binary arithmetic operator
(b) Postfix increment and Prefix increment
(c) Postfix decrement and Prefix decrement
(d) <code>(p! = q)</code> and <code>!(p = q)</code>



5. What is the difference between  
 (a) / and % operator?  
 (b) = and ==? [ICSE 2011]
6. What will be the output of the following code?  
 (a) 

```
int k=5,j=9;
k+= k++ - ++j + k;
System.out.println("k="+k);
System.out.println("j="+j);
```

 [ICSE 2011]  
 (b) If `int y = 10` then find `int z = (++y*(y+++5))`; [ICSE 2015]  
 (c) Give the output of the following expression: [ICSE 2016]  
`a+= a++ + ++a + --a + a--`; when `a = 7`;  
 (d) What is the value of `y` after the execution? [ICSE 2018]  
`y+= ++y + y-- + --y`; when `int y=8`
7. Rewrite the following program segment using if-else statements instead of the ternary operator.  
 (a) `String grade = (marks>=90)? "A": (mark>=80)? "B": "C";` [ICSE 2014]  
 (b) `commission = (sale > 5000) ? sale*10/100 : 0;`  
 (c) `net = (salary > 10000) ? salary - (8.33/100)*salary : salary - (5/100)*salary.`  
 (d) `s= (a+b < c || a+c <= b || b+c <= a) ? "Triangle is not possible": "Triangle is possible";`  
 (e) `c = (x >= 'A' && x <= 'Z') ? "Upper Case Letter" : "Lower Case Letter";`
8. Rewrite the following using ternary operator.  
 (a) 

```
if (x%2 == 0)
System.out.println("Even");
else
System.out.println("Odd");
```

 [ICSE 2016]  
 (b) 

```
if (bill>10000)
discount=bill*10.0/100;
else
discount=bill*5.0/100;
```

 [ICSE 2018]  
 (c) 

```
if(income< 10000)
tax = 0;
else
tax = 12;
```

  
 (d) 

```
if(a>b)
{
if(a>c)
g=a;
else
g=c;
}
if(b>c)
g=b;
else
g=c;
```

  
 (e) 

```
if(p>=4750)
k=p*5/100;
else
k=p*10/100;
```



## Unit V (a): Introduction to Java

### LEARNING SCOPE

- Basic features of Java
- Java Compiler and Interpreter
- Java Libraries in JDK
- Java Reserved Words
- Basic Elements in Java Programming
- Output Statement in Java Programming
- Java Programming using BlueJ and Execution.

Java is an object oriented programming language, developed by James Gosling along with his associates at Sun Micro System (a software company), USA. Initially it was named OAK (a tree outside Gosling's office). The name OAK remained only for some hours until the development team realised that another language is also available with this name. Finally, on the coffee table they decided a professional name for the language as Java. It was primarily developed to help users in web designing but later on it emerged as a full-fledged programming language in 1991.



Java gained wide popularity amongst the users due to its platform independence. The platform independence means the program in Java language can be executed on any computer.

Java language allows writing a program in two ways. They are:

- Java application or Standalone system

Java application is a Java program that is developed to run on a computer without any help of a web browser. It is also known as standalone system.

- Java applet or Internet applet

Java applet is a program written in Java language that can be included in a web page in the same way as a picture or a clip is included. Java applets are run with the help of a Java supported web browser.

### Features of Java Language

Java language has the following features:

1. Java is object oriented language.
2. It is platform independent.
3. It uses compiler as well as interpreter.
4. It is a robust language.
5. It is case sensitive. The upper case and lower case letters are distinguished by the language.



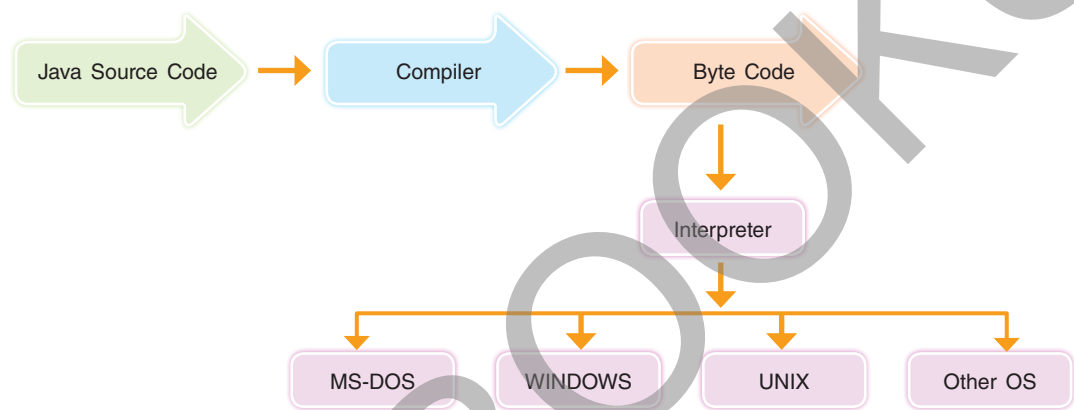


## Java Compiler and Interpreter

It is a well-known fact that a computer understands instructions only in machine language (i.e., binary language). The application programs are written in computer by using high-level languages. Hence, a program in high-level language needs to be converted into machine code for the computer to understand and execute.

Java compiler is a system software that converts a program in Java (Java source code) into an intermediate binary form called Byte Code. Further, the byte code is converted into machine code suitable to a specific platform known as Java Interpreter.

Java interpreter enables a computer to execute a Java program as well as a program written in other language compiled into Java byte code. This is the reason, Java interpreter is said to be Java Virtual Machine (JVM).



It is important to know that Java machine code varies for different platforms such as windows (single user OS), UNIX (multi-user OS) including other operating systems.

## Packages in JDK 1.5

As you know that Java Development Kit (JDK) 1.5 has a library called JCL (Java Class Library). It contains various packages. A package is a collection of classes such that each class includes different in-built functions. Some of the packages in JDK are listed below.

java.lang	Supports classes containing String/Character, Math, Integer, Thread, etc.
java.io	Contains classes for supporting input and output statements.
java.applet	Contains classes to generate applets.
java.net	Contains classes for supporting networking operations.
java.awt	Contains classes for supporting abstract window tool kit and managing GUI (Graphic User Interface) components.
java.text	Provides classes for handling formatting of text, dates, numbers and messages.
java.math	Contains classes for supporting mathematical functions such as square roots (integer and decimal both).

## Using a Package

During the course of programming, we often use functions of a class which belong to a specific package. It can be done by including the package in our program. A package can be included by using the keyword ‘import’ followed by the package name.



e.g.,

```
import java.io.*;
```

```
import java.util.*;
```

'\*' sign denotes that all the classes of the package will be imported in the program. If you want to use only a particular class of the package then mention its name in place of '\*' sign.

e.g., import java.util.Scanner;

A package that is imported by default is java.lang.

## Keyword or Reserved word

There are some words that carry special meaning to the system compiler. These words can't be used as variable names in the program. Such words are called as Keywords or Reserved Words.

Some of the keywords used in programming are listed below.

case	switch	else	break	static
do	const	throws	float	char
try	int	double	void	goto
for	while	new	import	boolean
long	if	byte	package	private
catch	short	public	class	default

## Basic elements of Java Programming

The basic format of writing a program in Java is entirely different from that of other languages. So, it is essential to know the following points while writing a Java program.

- **Class Declaration:** It is the first line of the program. It uses a keyword class followed by a name called class name.
- **Main function:** The main( ) function encloses the programming statements within opening and closing curly braces { }. As the main function is a part of class declaration, it is further enclosed within curly braces { } under class heading.

## Output Statement in Java Programming

The output statements in Java are used to display the result of the program on the screen. They are written in two ways:

- **System.out.print( ):** This statement is used to display the value enclosed within its braces. It leaves the cursor in the same line on the screen.

**Example:**

```
int p = 25, q = 20;
```

```
System.out.print("First number is" + p);
```

```
System.out.print("Second number is" + q);
```

```
System.out.print("The sum of two numbers is" + (p+q));
```

If a message is to be displayed along with a value then they must be separated by using + sign.

**Output:**

First number is 25 Second number is 20 The sum of two numbers is 45

- **System.out.println( ):** This statement displays the output of the program on the screen in different lines. The letters ‘\n’ in the statement act as a line feed which directs the cursor to move to the next line.

**Example:**

```
int p = 25, q = 20;
```

```
System.out.println("First number is" + p);
```

```
System.out.println("Second number is" + q);
```

```
System.out.println("The sum of two numbers is" + (p+q));
```

**Output:**

```
First number is 25
Second number is 20
The sum of two numbers is 45
```

## Java Programming using BlueJ

BlueJ version is Java platform for window based applications. It uses a menu driven approach that allows the users to carry out different tasks by clicking the options available on the menu bar. Like JDK 1.5, BlueJ also uses compiler as well as interpreter to convert source code into machine code. A sample program appears on the screen to provide assistance to new users. These features of BlueJ make it more convenient for the programmers in coding their programs.

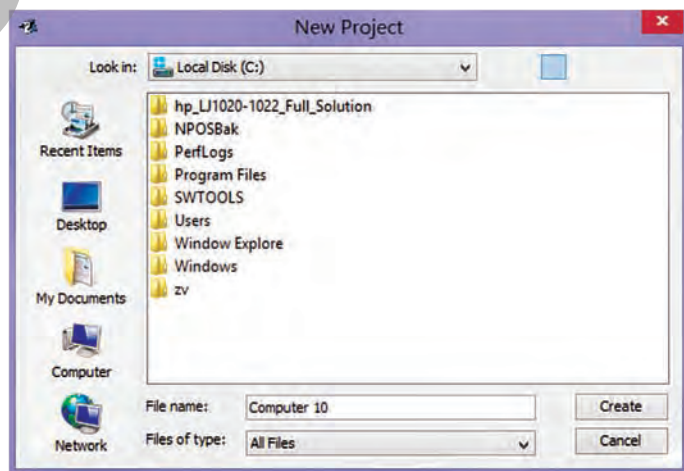
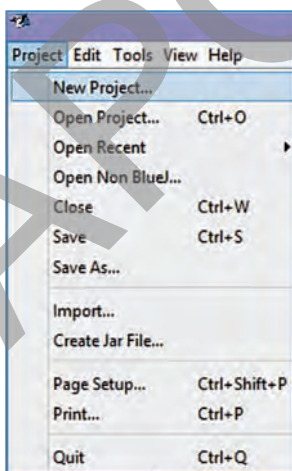
### Java Program on BlueJ Platform

The steps to write a first small program in BlueJ are as follows:

**Step 1 :** Double click on the BlueJ icon.

**Step 2 :** Click ‘Project’ and select ‘New Project’.

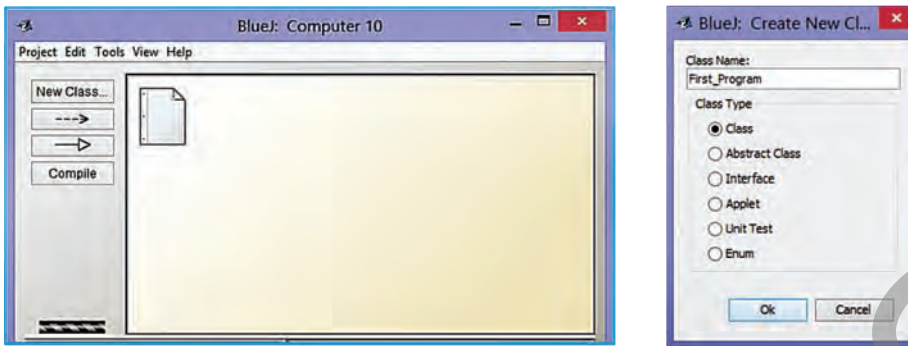
**Step 3 :** Enter the project name (say, Computer 10) in New Project window and click on ‘Create’ button.



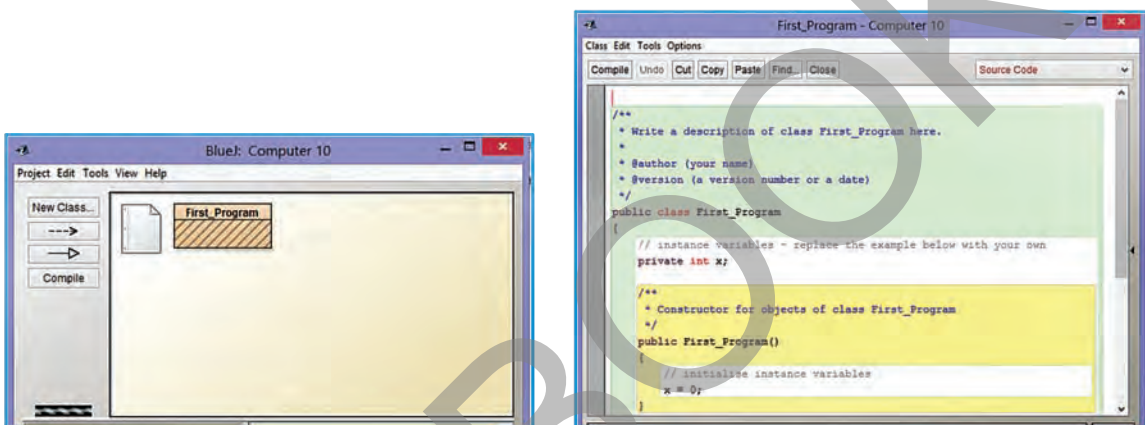
**Step 4 :** The project with the name ‘Computer 10’ appears on BlueJ window.

**Step 5 :** Click ‘New Class’ and enter a class name (say, First\_Program) in the Class Name text box.

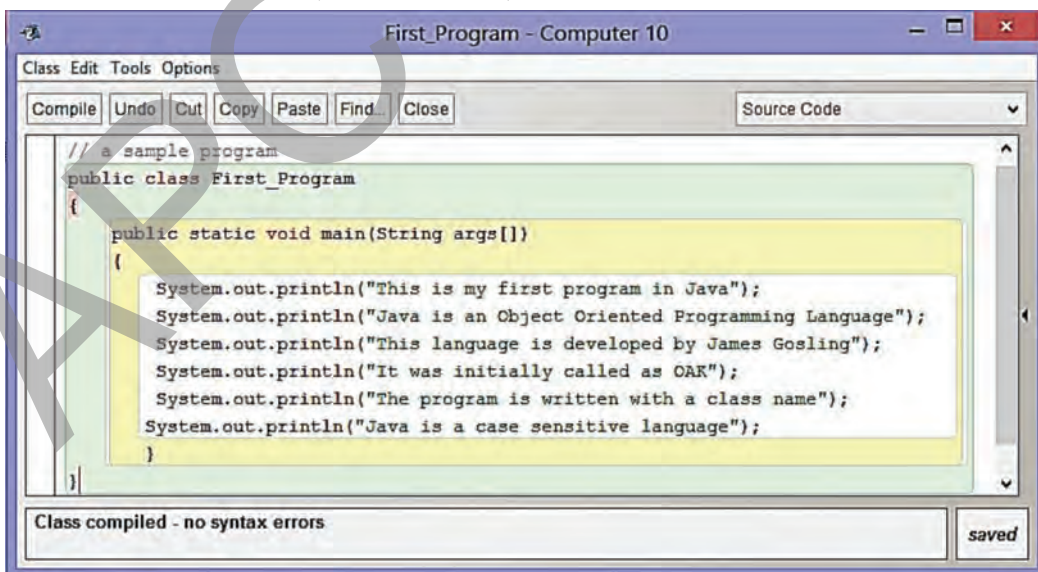
**Step 6 :** Finally, click ‘Ok’ to create class icon. It a program file in which the user writes program.



**Step 7 :** On double clicking class icon (First\_Program), a sample program is displayed. This program acts as a template to the user.



**Step 8 :** Erase the sample program and start writing the program on the blank screen. It is shown as below:



**Step 9 :** Click on ‘Compile’ button. If the program is error-free, a message will appear as ‘Class compiled – no syntax errors’. Thus, you are ready to execute your program.

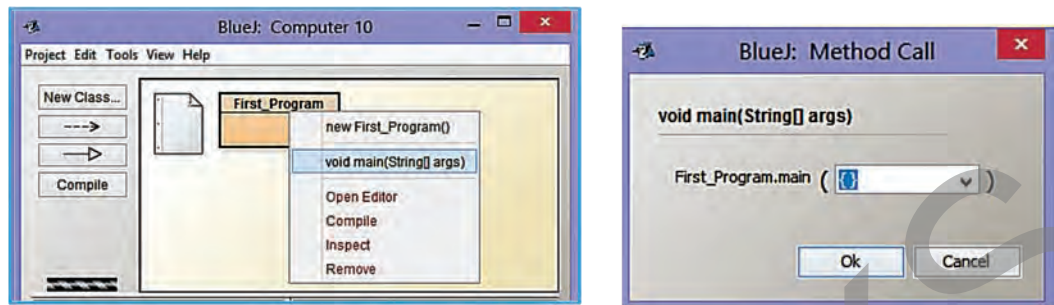




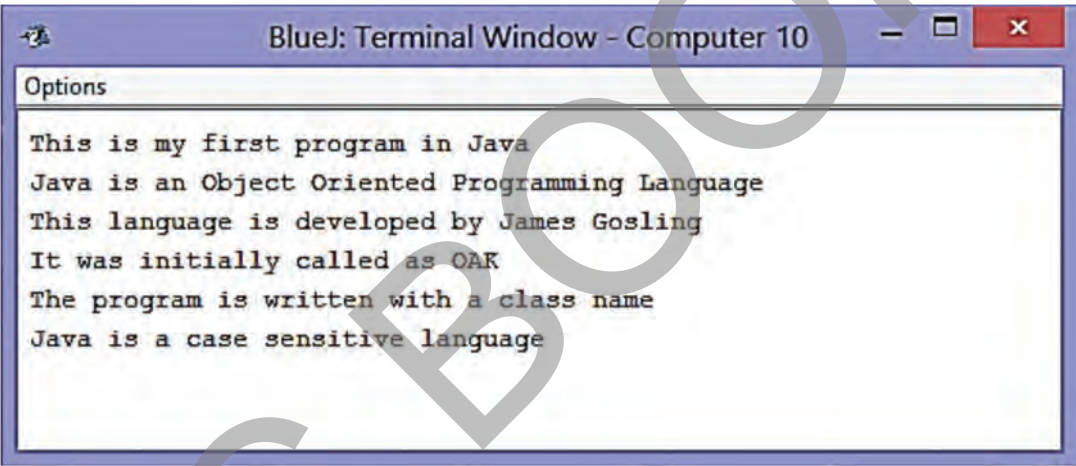
**Execution on BlueJ platform**

The different steps to execute the program are given below:

- Step 1** : Close the program window.
- Step 2** : Right click the class icon (First\_Program). A drop down menu appears.



- Step 3** : Select the option 'void main(String[ ] args)'.
- Step 4** : A 'Method Call' dialog box appears on the screen. Click Ok.
- Step 5** : The output will appear on the 'Terminal Window', i.e., the output screen.



- Step 6** : Finally, close the Terminal Window.



## REVIEW Insight

**Q.1** Name a menu based Java programming platform.

**Ans.** BlueJ

**Q.2** What is the another name of Java Interpreter?

**Ans.** JVM (Java Virtual Machine)

**Q.3** What does the term language processor mean?

**Ans.** Language processor is the term used for translators such as compiler, interpreter, etc.

**Q.4** Name the person who developed Java language.

**Ans.** James Gosling

**Q.5** Define the term byte code.

[ICSE 2010]

**Ans.** Java compiler converts source code to an intermediate binary code. This code helps generate machine code and is known as byte code.

**Q.6** What is meant by a package? Name any two Java application programming interface packages.

[ICSE 2014]

**Ans.** Packages contain logically related classes together. Each package defines a number of classes which can be used in the program by importing a single package.

Two Java application programming interface packages are:

1. java.io

2. java.util

**Q.7** Name the package that contains the class:

[ICSE 2015]

(i) BufferedReader

(ii) Scanner

**Ans.** (i) java.io.\*

(ii) java.util.\*

**Q.8** Name any two library packages.

[ICSE 2016]

**Ans.** (i) java.io.\*

(ii) java.util.\*

**Q.9** Write a statement in Java to display:

Best of Luck

for

ICSE 2020 Examination

**Ans.**

```
System.out.println("Best of Luck");
System.out.println("for");
System.out.println("ICSE 2020 Examination");
```

**Q.10** Given: 

```
System.out.print("BEST ");
```

[ICSE 2018]

```
System.out.println("OF LUCK");
```

Choose the correct option for the output of the above statements.

(i) BEST OF LUCK

(ii) BEST OF LUCK

**Ans.** (i) BEST OF LUCK

**Q.11** How does a byte code gets executed?

**Ans.** Java byte code is converted to machine code with the help of Java interpreter and gets executed on computer.

**Q.12** Write two features of Java language.

**Ans.** (i) Java language is platform independent.

(ii) Java language is case sensitive.



## EXERCISES

### I. Tick the correct answer:

1. A Java program developed and executed by the users without using web browser is known as:  
(a) application      (b) applet      (c) object      (d) none
2. Who developed Java?  
(a) James Gosling      (b) Robert James  
(c) Bjarne Stroustrup      (d) None
3. Java Virtual Machine (JVM) is an:  
(a) interpreter      (b) compiler  
(c) machine code      (d) byte code
4. Java is a case sensitive language. Which is the most appropriate statement with respect to this context?  
(a) Upper and lower case letters are distinguished.  
(b) Upper and lower case letters are ignored.  
(c) Only lower case letters are distinguished.  
(d) None.
5. Which of the following package is needed to find the square root of a number?  
(a) java.text      (b) java.math  
(c) java.lang      (d) None
6. Which of the following is not a Java reserved word?  
(a) private      (b) public  
(c) break      (d) total
7. The most suitable statement for BlueJ is:  
(a) It is a window based platform.  
(b) It is a DOS based platform.  
(c) It is a window based and DOS based platform.  
(d) None.

### II. Fill in the blanks:

1. The initial name of Java was .....
2. .... converts source code to byte code.
3. JVM is used to convert ..... into machine code.
4. Java is ..... independent language.
5. Windows based Java platform is known as .....
6. Reserved words are also called .....
7. .... package is used for mathematical functions in Java.

### III. Predict the output of the following snippets:

1. 

```
System.out.println("My name is ");
System.out.print("Kunal Kishore");
System.out.println("I am a student of Class X");
```





- ```
2. System.out.println("This is my first Java program");
   System.out.print("Display the message:");
   System.out.print("Have fun!!!");

3. System.out.println("India got independence");
   System.out.println("on");
   System.out.println("15th August,1947");

4. System.out.print("The factorial of 5 is");
   System.out.println("120");
```

#### IV. Write short answers:

- [illegible]

## Unit V (b): Input in Java

### LEARNING SCOPE

- Input by using Function Argument
- Input by using Stream Class
- Input by using Scanner Class
- Input using Command Line Argument
- Types of Errors
- Syntax Error
- Logical Error
- Run Time Error
- Comment Statements in Java.

Input means to provide data to a computer for processing. While writing a program, we use input statement to accept data from the user. Hence, input statement enables user to enter data at run time (i.e., during execution of a program).

The different computer languages have different syntax to input values. Java language allows users to enter data for processing in the following ways:

1. Using Function Argument
2. Using InputStreamReader Class
3. Using Scanner Class
4. Using Command Line Argument

### 1. Using Function Argument

This is one of the methods to accept the value from the user at the time of execution of the program. The variables whose values are to be input must be provided as arguments to the main( ) function.

**Example:** `public static void main(int a, int b)`

The above function will accept two integer numbers from the user. The output of the program is obtained on the screen after execution. Let us have a look on the sample programs.

### Sample Programs:

**Prog. 1** Write a program to input principal, rate and time. Calculate and display the difference between Simple Interest (SI) and Compound Interest (CI) by using function argument.

[Hint:  $CI = A - P$ ; where,  $A = P \left( 1 + \frac{R}{100} \right)^T$ ]

//To find the difference between Compound Interest and Simple Interest

```
public class Difference
```

```
{
```

```
public static void main(int p, int r, int t)
```

```
{
```

```
double si,ci=0,amt,diff=0;
```

```
si=p*r*t/100.0;
```

```
amt=p*(Math.pow(1+r/100.0,t));
```

```
ci=amt-p;
```

```
diff=ci-si;
```

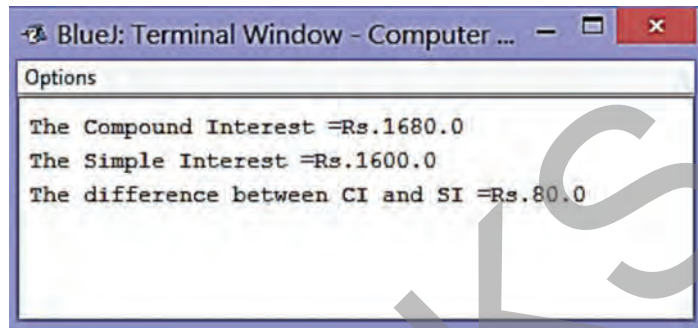
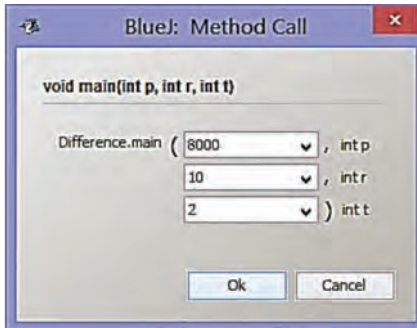
```
System.out.println("The Compound Interest =Rs."+(float)ci);
```

```

System.out.println("The Simple Interest =Rs."+si);
System.out.println("The difference between CI and SI =Rs."+(float)diff);
}
}

```

## Execution of the program



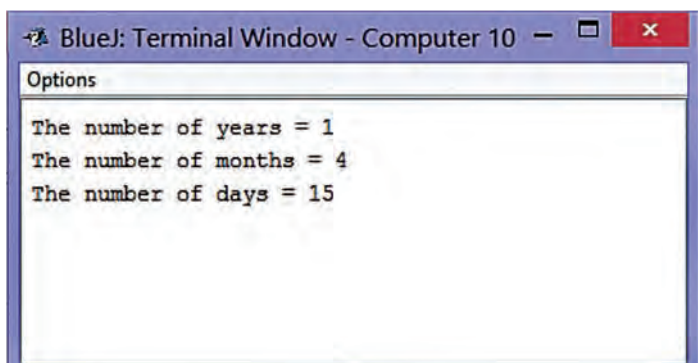
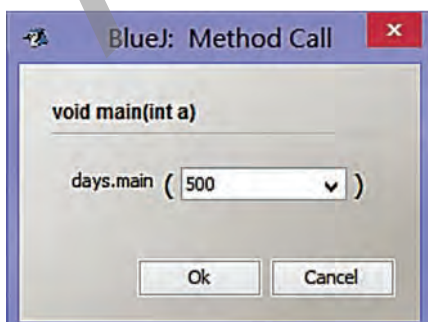
**Prog. 2** Write a program in Java to accept the number of days and display the result after converting it into number of years, number of months and the remaining number of days.

```

// To convert days into years, months and days
public class days
{
    public static void main(int a) //using function argument
    {
        int b,y,m,d;
        y=a/365;
        b=a%365;
        m=b/30;
        d=b%30;
        System.out.println("The number of years = "+ y);
        System.out.println("The number of months = "+ m);
        System.out.println("The number of days = "+ d);
    }
}

```

### Output:



## 2. Using Stream Class

To input a value by using `InputStreamReader` class, use the following steps:

- (i) `InputStreamReader` class is available in `java.io` package. Hence, first of all we need to import this package by using the statement:

```
Import java.io.*
```

- (ii) In the main function, create objects of `InputStreamReader` and `BufferedReader` classes as shown below:

```
InputStreamReader read = new InputStreamReader(System.in);
```

```
BufferedReader in = new BufferedReader(read);
```

The object names like *read* and *in* can be any word as per the user's choice.

The different types of values can be input in the following ways:

### ◆ Syntax to Accept Numeric Data Values

- To accept an integer:

```
int n;
```

```
System.out.println("Enter a number");
```

```
n = Integer.parseInt(in.readLine( ));
```

- To accept a decimal number:

```
float n;
```

```
System.out.println("Enter a decimal number");
```

```
n = Float.parseFloat(in.readLine( ));
```

- To accept a decimal fraction up to 64 bits:

```
double n;
```

```
System.out.println("Enter a decimal number");
```

```
n = Double.parseDouble(in.readLine( ));
```

#### Note:

While accepting data through `InputStreamReader` class and `Scanner` class, a relevant message must be displayed before the input statement. This is used as a prompt for the user and it also activates the input editor.

### ◆ Syntax to Accept a Character

```
char ch;
```

```
System.out.println("Enter a character");
```

```
ch = (char)(in.read( ));
```

### ◆ Syntax to Accept a String

```
String str;
```

```
System.out.println("Enter a String");
```

```
str = in.readLine( );
```

## A Sample Program

### Prog. 3

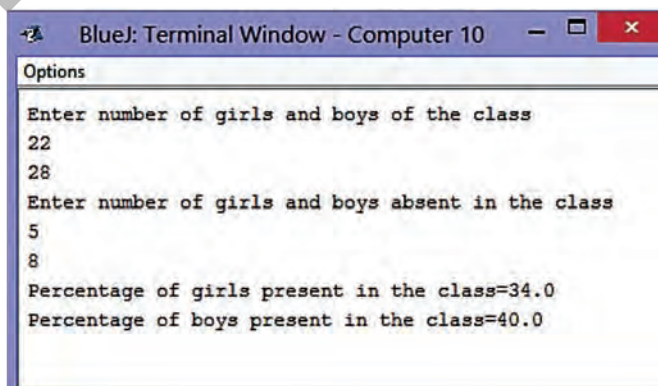
A class contains 'b' number of boys and 'g' number of girls. On a rainy day, 'ba' number of boys and 'ga' number of girls are absent. Write a program to input the values of 'b', 'g', 'ba' and 'ga'. Calculate and display the following:

- (i) Percentage of girls present in the class.
- (ii) Percentage of boys present in the class.

[Percentage must be calculated on the total number of students.]

```
// To find the percentage of girls and boys present in the class
import java.io.*;
public class Present
{
    public static void main(String args[]) throws IOException
    {
        int g,b,ga,ba;
        float pg,pb;
        InputStreamReader read= new InputStreamReader(System.in);
        BufferedReader in= new BufferedReader(read);
        System.out.println("Enter number of girls and boys of the class");
        g = Integer.parseInt(in.readLine());
        b = Integer.parseInt(in.readLine());
        System.out.println("Enter number of girls and boys absent in the class");
        ga = Integer.parseInt(in.readLine());
        ba = Integer.parseInt(in.readLine());
        pg=(float)(g-ga)/(g+b)*100;
        pb=(float)(b-ba)/(g+b)*100;
        System.out.println("Percentage of girls present in the class="+pg);
        System.out.println("Percentage of boys present in the class="+pb);
    }
}
```

### Output:



The screenshot shows a terminal window titled "BlueJ: Terminal Window - Computer 10". The output of the program is as follows:

```
Options
Enter number of girls and boys of the class
22
28
Enter number of girls and boys absent in the class
5
8
Percentage of girls present in the class=34.0
Percentage of boys present in the class=40.0
```



### 3. Using Scanner Class

To input a value by using Scanner class use the following steps:

- (i) Scanner class is a member of java.util package. Hence, first of all, import java.util package in your program as shown below:

```
import java.util.*; or import java.util.Scanner;
```

↓

Package

↓

Dot

↓

Class

operator

- (ii) Create object of Scanner class in the main function as shown:

```
Scanner sc = new Scanner (System.in);
```

↓

Class

↓

Scanner

↓

Object

↓

New

↓

Operator

↓

Class

You can use any object name of your choice.

- (iii) The values of different data types can be input by using next() functions of Scanner class as per the table shown below:

| Types of data to be entered | Functions to enter data                                 |
|-----------------------------|---------------------------------------------------------|
| integer                     | int n = sc.nextInt( );                                  |
| float                       | float f = sc.nextFloat( );                              |
| double                      | double d = sc.nextDouble( );                            |
| string                      | String s = in.next( ); or<br>String s = in.nextLine( ); |

### A Sample Program

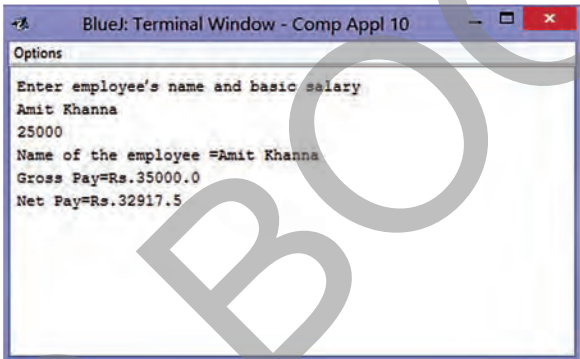
**Prog. 4**

Write a class with name employee and basic as its data member. Find the gross pay of an employee for the following allowances and deduction. Use meaningful variables:  
Dearness Allowance = 25% of Basic Pay  
House Rent Allowance = 15% of Basic Pay  
Provident Fund = 8.33% of Basic Pay  
Net Pay = Basic Pay + Dearness Allowance + House Rent Allowance  
Gross Pay = Net Pay – Provident Fund  
[ICSE 2005]

```
/** A program to calculate Employee's Gross and Net Salary by using Scanner Class*/
import java.util.*;
public class Employee_Salary
{
public static void main(String args[])
{
Scanner in = new Scanner(System.in);
```

```
System.out.println("Enter employee's name and basic salary");
double basic,da,hra,pf,gp=0,np=0;
String empn;
empn=in.nextLine();
basic= in.nextInt();
da=basic*25.0/100.0;
hra=basic*15.0/100.0;
pf=basic*8.33/100.0;
gp=basic+da+hra;
np=gp-pf;
System.out.println("Name of the employee =" +empn);
System.out.println("Gross Pay=Rs." +gp);
System.out.println("Net Pay=Rs." +np);
}
}
```

Output:



4. Using Command Line Argument

This is one of the ways to accept the data values from the user and pass the arguments (known as command line arguments) to the main function. While accepting the values from the console, the system stores the data values in different locations as an array of strings. The arguments to the main function are passed through args[0], args[1], args[2] and so on.

To input a value by using command line arguments use the following steps:

- (i) Use a string type array (args[]) as an argument to the main function as shown below:  
`public static void main(String args[ ]) or (String [ ]args)`
- (ii) The values of different data types can be input as per the table shown below:

| Types of data to be entered | Functions to input data                              |
|-----------------------------|------------------------------------------------------|
| integer                     | <code>int n = Integer.parseInt(args[0]);</code>      |
| float                       | <code>float f = Float.parseFloat(args[0]);</code>    |
| double                      | <code>double d = Double.parseDouble(args[0]);</code> |
| string                      | <code>String s = args[0];</code>                     |





## A Sample Program

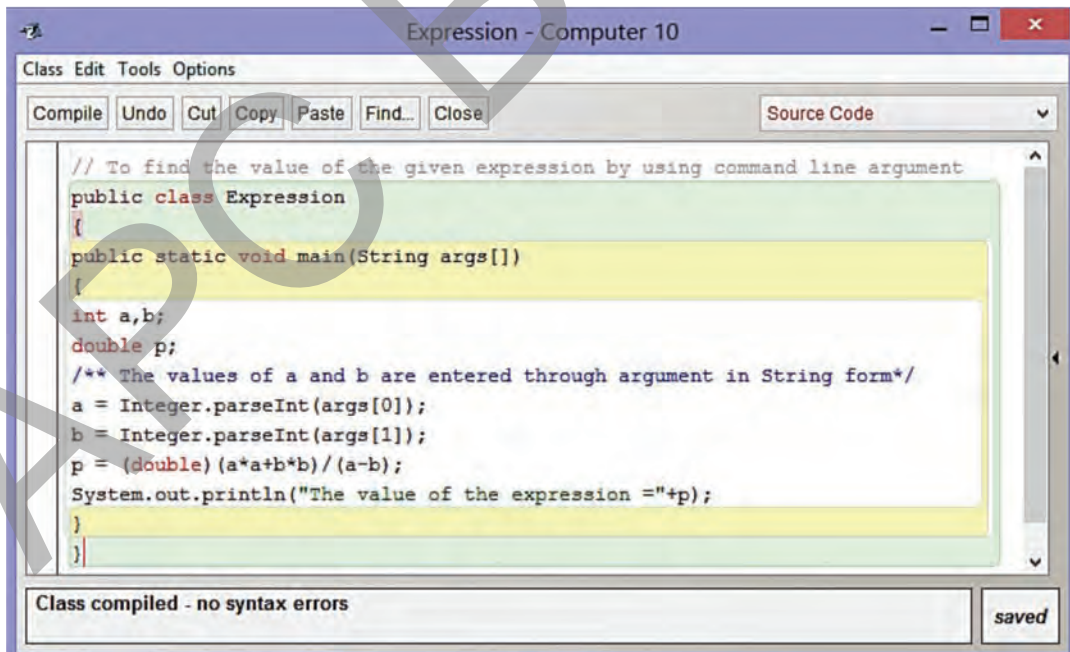
**Prog. 5** Write a program in Java to find the value of the given expression.

$$p = \frac{a^2 + b^2}{(a - b)}$$

```
// To find the value of the given expression by using command line argument
public class Expression
{
public static void main(String args[])
{
int a,b;
double p;
/** The values of a and b are entered through argument in String form*/
a = Integer.parseInt(args[0]);
b = Integer.parseInt(args[1]);
p = (double)(a*a+b*b)/(a-b);
System.out.println("The value of the expression =" +p);
}
}
```

### Compilation and Execution

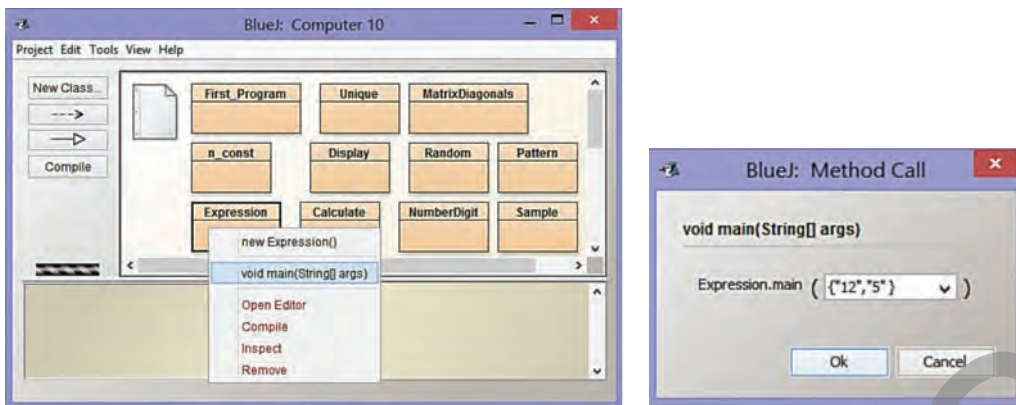
**Step 1** : Compile and close the program.



**Step 2** : Select the folder 'Expression'.

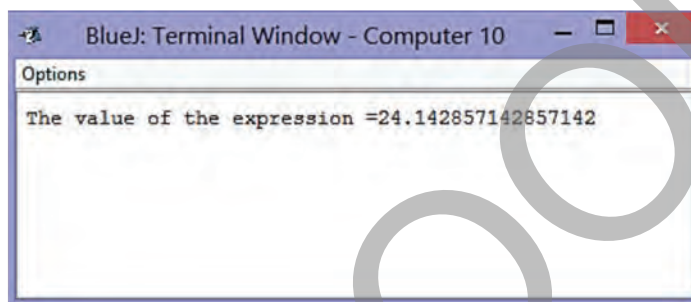
**Step 3** : Select void main(String[ ] args) from dropdown list.

**Step 4** : A Method Call window will appear on the screen, click Ok.



**Step 5 :** Enter the value as ‘Strings’ in the space provided under the ‘Expression.main( )’ in the Method Call window (as shown).

**Step 6 :** Click Ok.



The output of program will appear on the Terminal Window.

## Types of Errors

When you run a program then it may happen that you may not get the desired result due to incorrect input data and hence, some error arises. Generally, there are three types of errors that can occur in a computer program, i.e.,

1. Syntax error
2. Logical error
3. Run time error

### 1. Syntax error

These errors occur due to the grammatical errors in the programming language. It may be missing of punctuation sign, incorrect instructions, undefined variables, etc.

A snippet is illustrated below:

```
int a=145,b=40,c=50;
p= (a+b)/c;
System.out.println("The value of the expression =" +p);
```

Here, you can notice that the data type of variable ‘p’ is not defined in the program. Thus, the program shows a syntax error. If you define the data type of the variable ‘p’ as double, it shows ‘No syntax error’.

Thus, the program is error-free and ready to execute.



## 2. Logical error

The error which occurs in the programming logic is known as logical error. In this case, the program is compiled successfully but does not produce the desired results.

A snippet is illustrated below:

```
int a=160,b=40,c=50;
float p=0;
p= a+b/c;
System.out.println("The value of the expression =" +p);
```

Ans. = 160.8

In the snippet shown above, there is no grammatical error in the structure of the program. Hence, program will compile and execute the result. But the result is not the desired result as per the requirement.

Here, user wants to divide the sum of 'a' and 'b' by c. But the statement  $p = a+b/c$  instructs the compiler to add 'a' with the result of  $\frac{b}{c}$ , which is a logical error. To overcome this error, the statement should be written as shown below:

```
int a=160,b=40,c=50;
float p=0;
p= (a+b)/c;
System.out.println("The value of the expression =" +p);
```

Ans. = 4.0

Thus, the correct result is obtained.

## 3. Run Time error

It is a type of error that occurs at runtime when the compiler does not respond properly while executing a statement.

*For example:*

Dividing a number by zero or trying to find out the square root of a negative number, etc.

Consider the following snippet.

```
int a=160,b=40,c=0;
float p=0;
p= (a+b)/c;
System.out.println("The value of the expression =" +p);
```

Here, the value of  $c = 0$ , so the result will come to an infinite value, which is a run time error.



## Comment Statements in Java Programming

Comments in a program make it more readable for the user. They are not executed by the compiler. They only provide information about the code to the user.

There are three ways to give a comment in Java programming. They are:

1. Single line comment (//)
2. Multi line comment (/\* ..... \*/)
3. Documentation comment (/\*\* ..... \*/)

### 1. Single Line Comment

It is a single line comment which provides short relevant information about the program. A single line comment starts with // followed by the message.

**For example:** // A program to find the sum of two numbers

### 2. Multi line Comment

It is a way to present more information about programs to explain the logic in detail. This comment starts with /\* followed by the comment lines and ends with \*/.

**For example:** /\* A program to calculate discount  
It accepts the marked price and discount percent as inputs\*/

### 3. Documentation Comment

Documentation comment is not a discussion about the program logic but includes some text document to be viewed by other users. This comment starts with /\*\* followed by the set of comment lines and ends with \*/.

**For example:** /\*\* Author name : Amit Shukla  
Starting date of project: 05/03/2018  
Completion date of project: 12/05/2018\*/

## REVIEW Insight

**Q.1** What is the purpose of the following functions?

- (i) nextInt( )
- (ii) nextLine( )

**Ans.** (i) nextInt( ) function is used to accept an integer value from the user at the time of execution.

(ii) nextLine( ) function accepts a string including spaces at the time of execution.

**Q.2** Predict the output in each case:

- (i) Math.sqrt (-36)
- (ii) Math.pow(-4,3);

**Ans.** (i) The square root of a negative number can't be determined. Hence, the output will be NaN.

(ii) The output: -64



- Q.3** Why do you need to mention the keyword `import` in the program?
- Ans.** The keyword `import` allows to include a package in the program.
- Q.4** What happens when
- (i) the value of a variable is divided by zero?
  - (ii) semicolon is missing in an input/output statement?
- Ans.** (i) When a value is divided by zero, 'Run Time' error occurs.  
(ii) If semicolon is missing in the input/output statement, then 'Syntax' error occurs.
- Q.5** Name the method of scanner class that is used to input:
- (i) a decimal number from the console.
  - (ii) a word or text of a line from the keyboard.
- Ans.** (i) `nextFloat( )` or `nextDouble( )`  
(ii) `next( )` or `nextLine( )`

## EXERCISES

### I. Fill in the blanks:

1. The package needed to import `StreamReader` class is .....
2. .... function accepts an integer by using scanner object.
3. The package needed to import `Scanner` class is .....
4. The keyword ..... is used to include a package in a program.
5. Division by a variable that contains a value zero is called ..... error.
6. .... statement is used to accept value at run time.
7. .... function accepts a string by using scanner object.
8. .... keyword refers to a non-returnable type function.
9. `Scanner ob = ..... Scanner(.....)`.
10. The command line argument accepts the data value as an ..... of Strings.

### II. Explain the following functions:

1. `Integer.parseInt(in.readLine( ));`
2. `(char)(in.read( ));`
3. `next( );`
4. `nextInt( );`
5. `nextLine( );`

### III. Answer the following:

1. What are the different ways to give input in a Java Program?
2. What is a package? Give an example.
3. What is the use of the keyword 'import' in Java programming?
4. What is a Run Time error? Explain with an example.
5. What are the different types of errors that take place during the execution of a program? Name them.
6. Give two differences between Syntax error and Logical error.

#### IV. Unsolved Java Programming:

1. In an election, there are candidates X and Y. On the election day, 80% of the voters go for polling, out of which 60% vote for X. Write a program to take the number of voters as input and calculate:
  - (i) number of votes X get
  - (ii) number of votes Y get
2. A shopkeeper offers 10% discount on the printed price of a mobile phone. However, a customer has to pay 9% GST on the remaining amount. Write a program in Java to calculate the amount to be paid by the customer taking printed price as an input.
3. A man spends  $\frac{1}{2}$  of his salary on food,  $\frac{1}{15}$  on rent,  $\frac{1}{10}$  on miscellaneous. Rest of the salary is his saving. Write a program to calculate and display the following:
  - (i) money spent on food
  - (ii) money spent on rent
  - (iii) money spent on miscellaneous
  - (iv) money savedTake the salary as an input.
4. Write a program to input time in seconds. Display the time after converting them into hours, minutes and seconds.  
Sample Input: Time in seconds: 5420  
Sample Output: 1 Hour 30 Minutes 20 Seconds
5. The driver took a drive to a town 240 km at a speed of 60 km/h. Later in the evening, he drove back at 20 km/h less than the usual speed. Write a program to calculate:
  - (i) the total time taken by the driver
  - (ii) the average speed during the whole journey

[Hint: average speed =  $\frac{\text{total distance}}{\text{total time}}$ ]
6. Write a program to input two unequal numbers. Display the numbers after swapping their values in the variables without using a third variable.  
Sample Input: a = 76, b = 65  
Sample Output: a = 65, b = 76
7. A certain amount of money is invested for 3 years at the rate of 6%, 8% and 10% per annum compounded annually. Write a program to calculate:
  - (i) the amount after 3 years.
  - (ii) the compound interest after 3 years.Accept certain amount of money (Principal) as an input.  
**Hint:**  $A = P \left(1 + \frac{R1}{100}\right)^T * \left(1 + \frac{R2}{100}\right)^T * \left(1 + \frac{R3}{100}\right)^T$  and  $CI = A - P$
8. The co-ordinates of two points A and B on a straight line are given as (x1,y1) and (x2,y2). Write a program to calculate the slope (m) of the line by using formula:  
$$\text{Slope} = \frac{(y2 - y1)}{(x2 - x1)}$$
Take the co-ordinates (x1,y1) and (x2,y2) as input.





## Unit VI: Mathematical Library Methods

### LEARNING SCOPE

- Methods of Math class viz. Math.min(a,b), Math.max(a,b), Math.pow(x,y), Math.sqrt(x), Math.cbrt(x), Math.log(x), Math.abs(x)
- Behaviour of positive and negative numbers in the functions such as Math.ceil(x), Math.floor(x), Math.round(x), Math rint(x)
- Math.exp()
- Trigonometrical functions
- Math.random( )

The library methods (functions) are the in-built methods designed by the developers. These predefined functions help the user to perform certain tasks easily and quickly which are frequently used in Java programming. In this unit, you will learn some mathematical functions which are included in a class called 'Math' under 'Java.lang' package.

### Methods of Math Class

The general syntax to use any method of Math class in Java program is as follows:

Math.<method name>

Let us recapitulate some methods frequently used in Java programming.

#### ◆ Math.min( )

It returns the minimum of two numbers. It returns int/long/double type value depending upon the type of arguments passed to it.

**Syntax:** <Return data type><variable> =<Function name(argument 1, argument 2)>;

e.g., double m= Math.min(-4.5, -5.4);

Output: -5.4

#### ◆ Math.max( )

It returns the greatest value of two given arguments. It returns int/long/double type value depending upon the type of arguments passed to it.

**Syntax:** <Return data type><variable> =<Function name(argument 1, argument 2)>;

e.g., double m= Math.max(9.2, 9.45);

Output: 9.45

#### ◆ Math.pow( )

This method is used to find the power raised to a given base value. It always returns a double type value.

**Syntax:** <Return data type><variable> =<Function name(argument 1, argument 2)>;

e.g., double m= Math.pow(2.0, 4.0);

Output: 16.0

#### ◆ Math.sqrt( )

It is used to find the square root of a positive number. It returns a double type value.

**Syntax:** <Return data type><variable> =<Function name(Positive number)>;

e.g., double m= Math.sqrt(7.29);

Output: 2.7

Square root of a negative number is an imaginary number.



### ◆ **Math.cbrt( )**

It is used to find the cube root of a positive or a negative number. It always returns a value in double data type.

**Syntax:** <Return data type><variable> =<Function name(Positive number)>;

e.g., double m= Math.cbrt(29.791);

Output: 3.1

### ◆ **Math.log( )**

The Math.log( ) function is used to find the natural logarithmic value of a given argument. It always returns a double type value.

**Syntax:** <Return data type><variable> =<Function name(Positive number)>;

e.g., double m= Math.log(6.25);

Output: 1.8325

### ◆ **Math.abs( )**

It always returns the absolute value of an argument (magnitude of the number without its sign, i.e., a positive value). The return value may be int/long/double data type depending upon the input arguments.

**Syntax:** <Return data type><variable> =<Function name(number)>;

e.g., double m= Math.abs(-12.45);

Output: 12.45

### ◆ **Math.round( )**

This method returns the value of a number rounded to its nearest integer. If the fractional part is less than 0.5 then it returns the same integer value, otherwise it returns the next higher integer. It returns the value in long or int type. It is obvious that it returns different output for positive and negative numbers.

**Syntax:** <Return data type><variable>=<Function name(argument)>;

- For positive numbers:

e.g., double m=Math.round(8.5);

Output: 9

- For negative numbers:

e.g., double m=Math.round(-8.5);

Output: -8

### ◆ **Math rint(-)**

This function returns the nearest integer of a given fractional number. The return data type will always be double.

**Syntax:** <Return data type><variable>=<Function name(argument)>;

- For positive numbers:

e.g., double m=Math.rint(8.5);

Output: 8.0

- For negative numbers:

e.g., double m=Math.rint(-9.5);

Output: -10.0



### ◆ **Math.ceil( )**

It returns the next higher integer number that is greater than or equal to the argument. It always returns the value as a double data type.

**Syntax:** <Return data type><variable>=<Function name(argument)>;

- For positive numbers:  
e.g., double m=Math.ceil(8.5);  
Output: 9.0
- For negative numbers:  
e.g., double m=Math.ceil(-8.5);  
Output: -8.0

### ◆ **Math.floor( )**

It returns the lower integer number that is less than or equal to the argument. It always returns the value as a double data type.

**Syntax:** <Return data type><variable>=<Function name(argument)>;

- For positive numbers:  
e.g., double m=Math.floor(8.912);  
Output: 8.0
- For negative numbers:  
e.g., double m=Math.floor(-8.912);  
Output: -9.0

### ◆ **Math.exp( )**

It results in the exponential value of an argument x (i.e.  $e^x$ ). It returns a double type value.

**Syntax:** <Return data type><variable> = <Function name(number)>;

e.g., double m=Math.exp(6.25);  
Output: 518.0128

### ◆ **Trigonometrical Functions**

There are some trigonometrical functions which are frequently used in mathematics to find the sine, cosine and tangent values respectively of a given angle (in radians). Angles are passed as an argument to the function. Some of these functions are:

- i. Math.sin( )                      ii. Math.cos( )                      iii. Math.tan( )

**Syntax:** <Return data type><variable> = <Function name(value in radian)>;

For example:

double d = Math.sin(x);

double d = Math.cos(x);

double d = Math.tan(x);

Here, the argument x is an angle whose values are assigned in radians. These functions return double type value.

Generally, the angles are measured in degrees. Thus, it is necessary to convert them into radian to use them in above functions.



## Conversion from degree to radian

To convert an angle from degree to radian, use the formula given as:

$$180 \text{ Degree} = \pi \text{ Radian}$$

$$\text{i.e., Degree} = \frac{22}{7} * \text{Radian} * \frac{1}{180}$$

Suppose, you want to calculate the value of  $\sin 30^\circ$ .

Let, int a = 30;

Then in radian:

$$x = 22/(7*180)*a;$$

$$\text{double } d = \text{Math.sin}(x);$$

Thus, it returns the value of  $\sin 30^\circ$  as 0.50.

Similarly,  $\text{Math.cos}( )$  and  $\text{Math.tan}( )$  can be calculated.

### ◆ **Math.random( )**

It returns a random number between 0 and 1 in a double data type value.

Syntax: <Return data type><variable> = <Function name( )>;

For example:

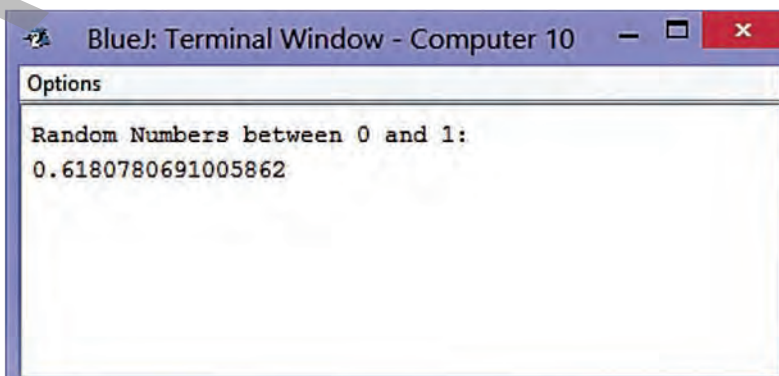
$$\text{double } d = \text{Math.random}( );$$

It will return any random double value for d between 0 and 1.

### **A Sample Program:**

```
// To display Random Number between 0 and 1
public class Random
{
    public static void main(String args[])
    {
        double d;
        System.out.println("Random Number between 0 and 1:");
        d=Math.random();
        System.out.println(d);
    }
}
```

### **The output:**



```
BlueJ: Terminal Window - Computer 10
Options
Random Numbers between 0 and 1:
0.6180780691005862
```



### Some facts about random function

- (i) To get an integer random number between 1 and n:  
`int r = (int) (Math.random() * n) + 1;` where n is the upper limit of the integer
- (ii) To get an integer random number between m and n where m and n are the lower and upper limits of the integer numbers respectively:  
`int r = (int) ((Math.random() * (n - m)) + m)`

#### Special cases:

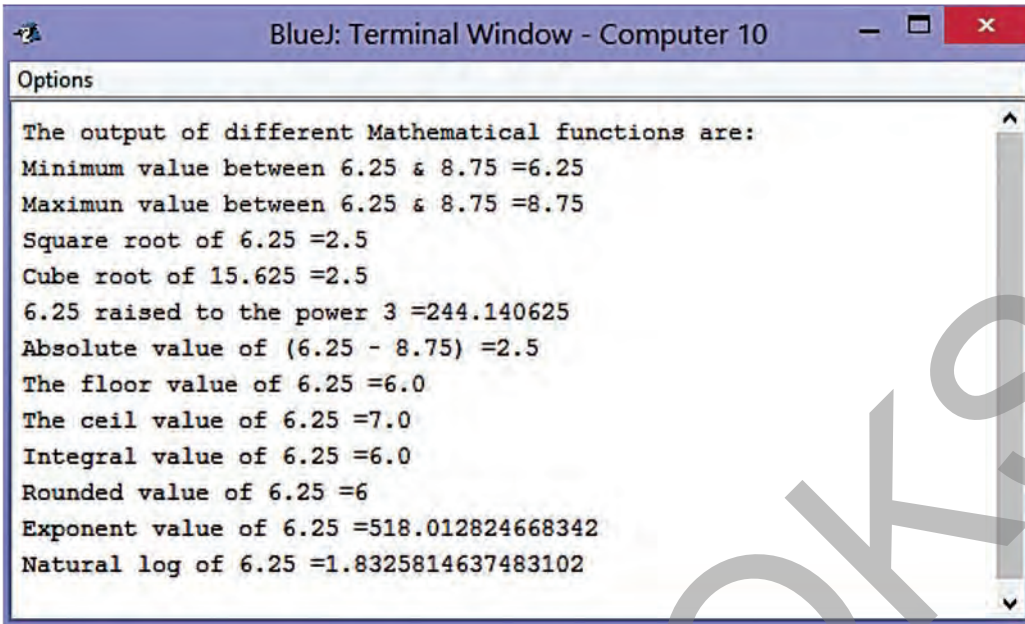
- (i) When `Math.random()` gives its lowest value as 0.1 then the lowest integer value of the range becomes zero (0).
- (ii) When `Math.random()` gives its highest value as 0.9 then the highest integer value of the range becomes 1.
- (iii) The random function can be used to obtain the results 'head' or 'tail' randomly when a coin is tossed. The function can be used as: `int n = (int)(Math.random() * 2);`  
Here, the value of n (i.e., 0 or 1) so obtained can be assumed as 'Tail' or 'Head' respectively.

### A single example using different mathematical functions

// A program to show the use of different mathematical functions

```
public class Maths_Functions
{
    public static void main(String args[])
    {
        double m=6.25, n=8.75, p=15.625;
        System.out.println("The output of different mathematical functions are:");
        System.out.println("Minimum value between 6.25 & 8.75 =" + Math.min(m,n));
        System.out.println("Maximun value between 6.25 & 8.75 =" + Math.max(m,n));
        System.out.println("Square root of 6.25 =" + Math.sqrt(m));
        System.out.println("Cube root of 15.625 =" + Math.cbrt(p));
        System.out.println("6.25 raised to the power 3 =" + Math.pow(m,3));
        System.out.println("Absolute value of (6.25 - 8.75) =" + Math.abs(m-n));
        System.out.println("The floor value of 6.25 =" + Math.floor(m));
        System.out.println("The ceil value of 6.25 =" + Math.ceil(m));
        System.out.println("Integral value of 6.25 =" + Math rint(m));
        System.out.println("Rounded value of 6.25 =" + Math.round(m));
        System.out.println("Exponent value of 6.25 =" + Math.exp(m));
        System.out.println("Natural log of 6.25 =" + Math.log(m));
    }
}
```

## Output of the program



```
Options
The output of different Mathematical functions are:
Minimum value between 6.25 & 8.75 =6.25
Maximun value between 6.25 & 8.75 =8.75
Square root of 6.25 =2.5
Cube root of 15.625 =2.5
6.25 raised to the power 3 =244.140625
Absolute value of (6.25 - 8.75) =2.5
The floor value of 6.25 =6.0
The ceil value of 6.25 =7.0
Integral value of 6.25 =6.0
Rounded value of 6.25 =6
Exponent value of 6.25 =518.012824668342
Natural log of 6.25 =1.8325814637483102
```

### SOLVED PROGRAMS ON MATHEMATICAL FUNCTIONS

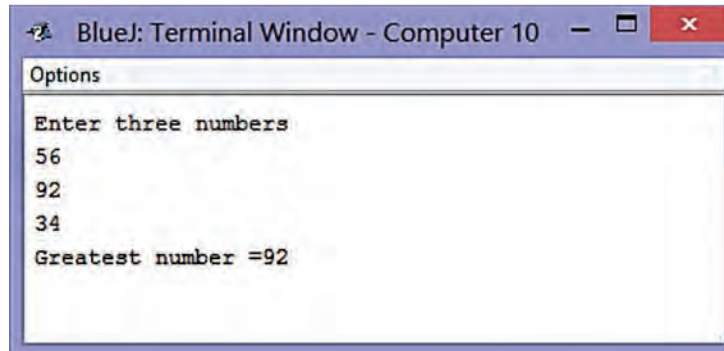
**Prog. 1** Write a program in Java to input three numbers and display the greatest number.

**Hint:** Use Math.max()

```
// To find the greatest number
import java.util.*;
public class Greatest
{
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
        int a,b,c,d,p;
        System.out.println("Enter three numbers");
        a=in.nextInt();
        b=in.nextInt();
        c=in.nextInt();
        d=Math.max(a,b);
        p=Math.max(c,d);
        System.out.println("Greatest number =" +p);
    }
}
```



## Output of the program



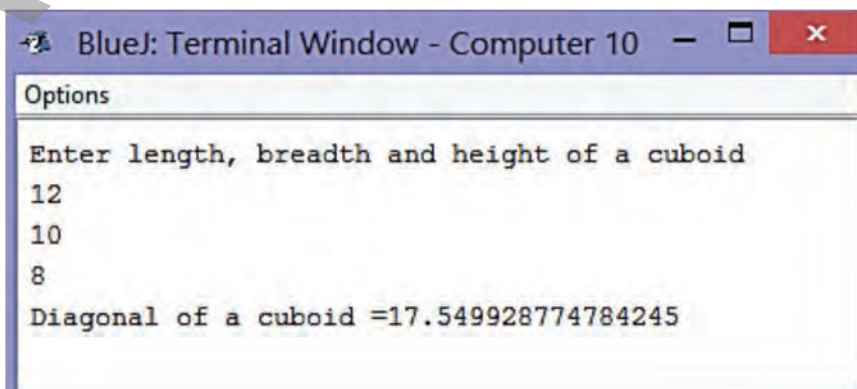
```
BlueJ: Terminal Window - Computer 10
Options
Enter three numbers
56
92
34
Greatest number =92
```

**Prog. 2** Write a program in Java to calculate the diagonal of a cuboid taking length, breadth and height as input.

**Hint:**  $\text{diagonal} = \sqrt{l^2 + b^2 + h^2}$

```
// To find the diagonal of a cuboid
import java.util.*;
public class Diagonal
{
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
        int l,b,h;
        double d;
        System.out.println("Enter length, breadth and height of a cuboid");
        l=in.nextInt();
        b=in.nextInt();
        h=in.nextInt();
        d=Math.sqrt(l*l+b*b+h*h);
        System.out.println("Diagonal of a cuboid =" +d);
    }
}
```

## Output of the program



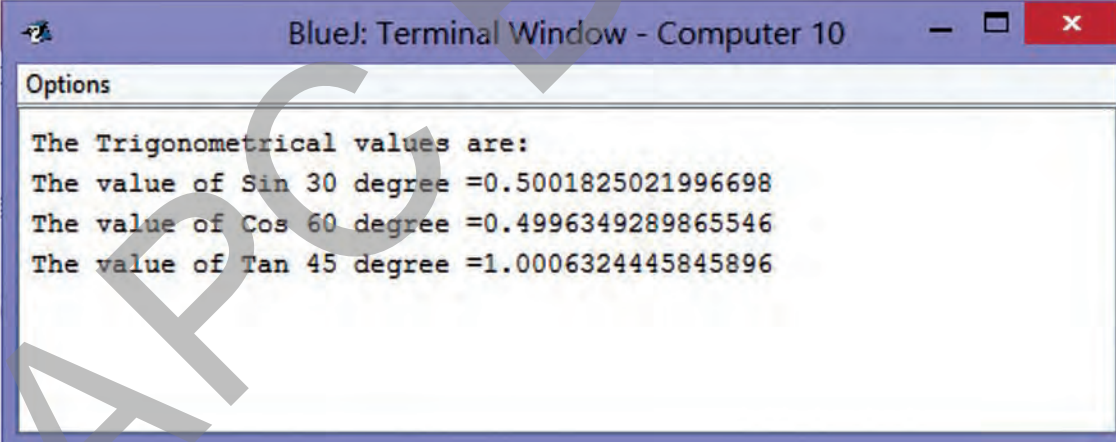
```
BlueJ: Terminal Window - Computer 10
Options
Enter length, breadth and height of a cuboid
12
10
8
Diagonal of a cuboid =17.549928774784245
```



**Prog. 3** Write a program to input three angles viz. 30, 45 and 60 (in degrees). Calculate and display the value of Sin 30°, Cos 60° and tan 45°.  
**Hint:** 180 degree =  $\frac{22}{7}$  radian

```
// To illustrate Trigonometrical functions
public class Trigonometry
{
public static void main(String args[])
{
double m=30,n=60,p=45;
double x,y,z;
// Convert angles in radians
x=(22.0/(7.0*180.0))*m;
y=(22.0/(7.0*180.0))*n;
z=(22.0/(7.0*180.0))*p;
System.out.println("The Trigonometrical values are:");
System.out.println("The value of Sin 30 degree =" + Math.sin(x));
System.out.println("The value of Cos 60 degree =" + Math.cos(y));
System.out.println("The value of Tan 45 degree =" + Math.tan(z));
}
}
```

### Output of the program



The screenshot shows a terminal window titled "BlueJ: Terminal Window - Computer 10". The output text is as follows:

```
Options
The Trigonometrical values are:
The value of Sin 30 degree =0.5001825021996698
The value of Cos 60 degree =0.4996349289865546
The value of Tan 45 degree =1.0006324445845896
```

**Prog. 4** Write a program to input a number and evaluate the results based on the number entered by the user:

- (a) Natural logarithm of the number
- (b) Absolute value of the number
- (c) Square root of the number
- (d) Cube of the number
- (e) Random numbers between 0 (zero) and 1 (one).

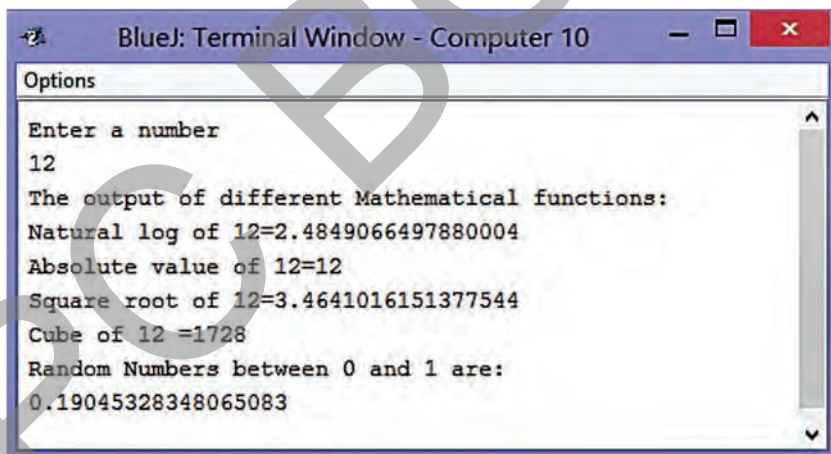
[ICSE 2006]





```
// To display the output of different mathematical functions
import java.util.*;
public class Results
{
public static void main(String args[])
{
Scanner in = new Scanner(System.in);
int a,i;
System.out.println("Enter a number");
a=in.nextInt();
System.out.println("The output of different Mathematical functions:");
System.out.println("Natural log of " + a + "=" + Math.log(a));
System.out.println("Absolute value of " + a + "=" + Math.abs(a));
System.out.println("Square root of " + a + "=" + Math.sqrt(a));
System.out.println("Cube of " + a + " =" + (a*a*a));
System.out.println("Random Numbers between 0 and 1 are:");
System.out.println(Math.random());
}
}
```

### Output of the program



The screenshot shows a terminal window titled "BlueJ: Terminal Window - Computer 10". The output of the program is as follows:

```
Options
Enter a number
12
The output of different Mathematical functions:
Natural log of 12=2.4849066497880004
Absolute value of 12=12
Square root of 12=3.4641016151377544
Cube of 12 =1728
Random Numbers between 0 and 1 are:
0.19045328348065083
```

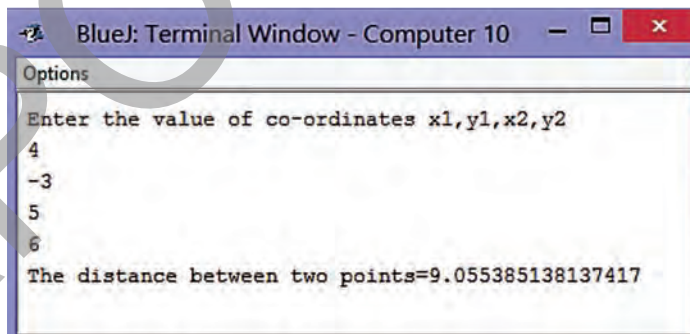
**Prog. 5**

The distance between two points A and B of a straight line, whose co-ordinates are (x1,y1) and (x2,y2), is calculated as:

$$AB = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}$$

Write a program to calculate the distance between the two points by taking the co-ordinates as inputs.

```
// To calculate the distance between the two points
import java.util.*;
public class Distance
{
public static void main(String args[])
{
Scanner in = new Scanner(System.in);
int x1,y1,x2,y2;
double p,d;
System.out.println("Enter the value of co-ordinates x1,y1,x2,y2");
x1=in.nextInt();
y1=in.nextInt();
x2=in.nextInt();
y2=in.nextInt();
p=Math.pow((y2-y1),2)+Math.pow((x2-x1),2);
d=Math.sqrt(p);
System.out.println("The distance between two points="+d);
}
}
```

**Output of the program**A screenshot of a terminal window titled "Blue: Terminal Window - Computer 10". The window shows the execution of the Java program. It prompts the user to "Enter the value of co-ordinates x1,y1,x2,y2". The user enters the values 4, -3, 5, and 6 on separate lines. The program then outputs "The distance between two points=9.055385138137417".

```
Blue: Terminal Window - Computer 10
Options
Enter the value of co-ordinates x1,y1,x2,y2
4
-3
5
6
The distance between two points=9.055385138137417
```



MATHEMATICAL FUNCTIONS AT A GLANCE

| Function                     | Description                                                           | Return data type          | Syntax                                       |
|------------------------------|-----------------------------------------------------------------------|---------------------------|----------------------------------------------|
| min(a,b)                     | Returns the smaller number between a and b.                           | int/long/float/<br>double | Math.min(a,b);                               |
| max(a,b)                     | Returns the greater number between a and b.                           | int/long/float/<br>double | Math.max(a,b);                               |
| sqrt(a)                      | Returns the square root of a positive number.                         | double                    | Math.sqrt(a);                                |
| cbrt(a)                      | Returns the cube root of a number.                                    | double                    | Math.cbrt(a);                                |
| pow(a,b)                     | Returns the value a <sup>b</sup> .                                    | double                    | Math.pow(a,b);                               |
| abs(a)                       | Returns the absolute value (magnitude) of any number.                 | int/long/float/<br>double | Math.abs(a);                                 |
| round(a)                     | Returns the rounded value up to the nearest integer.                  | int/long                  | Math.round(a);                               |
| floor(a)                     | Returns the rounded value down to the nearest integer.                | double                    | Math.floor(a);                               |
| ceil(a)                      | Returns the whole number greater than or equal to the number.         | double                    | Math.ceil(a);                                |
| rint(a)                      | Returns the truncated value of the number.                            | double                    | Math.rint(a);                                |
| random( )                    | Returns a random number between 0 and 1.                              | double                    | Math.random( );                              |
| log(a)                       | Returns the value of natural logarithm of a number.                   | double                    | Math.log(a);                                 |
| exp(a)                       | Returns an exponent value, i.e., e <sup>a</sup>                       | double                    | Math.exp(a);                                 |
| sin(a)/<br>cos(a)/<br>tan(a) | Returns the Sine/Cosine/Tangent of an angle a (expressed in radians). | double                    | Math.sin(a);<br>Math.cos(a);<br>Math.tan(a); |



**Q.1** What values will be stored in the variables p and q after executing the following codes?

- (i) `double a = -6.35; double b = 14.74;`  
`double p = Math.abs(Math.ceil(a));`  
`double q = Math rint(Math.max(a,b));`

**Ans.** p = 6.0  
q = 15.0

- (ii) `double b = -15.6;`  
`double p = Math.rint(Math.abs(b));`

**Ans.** p = 16.0

- (iii) `int p = Math.abs(Math.min(-61,-79));`  
`double q = Math.cbrt(4.913);`

**Ans.** p = 79  
q = 1.7

- (iv) `double a = -99.51, b = -56.25;`  
`double p = Math.abs(Math.floor(a));`  
`double q = Math.sqrt(Math.abs(b));`

**Ans.** p = 100.0  
q = 7.5

- (v) `double x=5.9, y=6.5;`  
`double p=(Math.min(Math.floor(x),y));`  
`double q=(Math.max(Math.ceil(x),y));`

**Ans.** 5.0  
6.5

**Q.2** What will the following functions return?

- (i) `Math.round(Math.pow(2.4,2));`  
(ii) `Math.abs(Math.max(-8,-11));`  
(iii) `Math.min(-23,-29);`  
(iv) `Math.ceil(14.8);`

**Ans.** (i) 6  
(ii) 8  
(iii) -29  
(iv) 15.0

**Q.3** What will the following functions display?

- (i) `System.out.println(Math.floor(-0.48));`  
(ii) `System.out.println(Math.ceil(-0.45));`  
(iii) `System.out.println(Math.ceil(8));`  
(iv) `System.out.println(Math.ceil(-8));`

**Ans.** (i) -1.0  
(ii) -0.0  
(iii) 8.0  
(iv) -8.0



## EXERCISES

### I. State whether the following statements are 'True' or 'False':

1. The return data type of `Math.log( )` is double.
2. `Java.Math` class is used for different mathematical functions.
3. The output of `Math.abs(-99.99)` is 100.00.
4. `Math.sqrt(-225)` can't be defined.
5. The output of `Math.cbrt(-3)` will be -27.
6. `Math.round( )` returns the value in float data type.

### II. Predict the output:

1. `System.out.println(Math.floor(-4.7));`

[ICSE 2018]

Ans. ....

2. `System.out.println(Math rint(-9.4)+Math.sqrt(9.0));`

Ans. ....

3. `System.out.println(Math.max(Math.ceil(14.55),15.5));`

Ans. ....

4. `System.out.println(Math.sqrt(Math.min(42.5,42.25)));`

Ans. ....

5. `System.out.println(Math.ceil(3.4)+Math.pow(2,3));`

[ICSE 2018]

Ans. ....

### III. Write down the syntax for the following functions:

1. To find the natural log of q.

Ans. ....

2. To find the cube root of a number m.

Ans. ....

3. To find the round-off of a number k.

Ans. ....

4. To find the absolute value of a number y.

Ans. ....

5. To find the exponent of a number p.

Ans. ....

### IV. Explain the following functions:

1. `Math.random( )`
2. `Math.pow( )`
3. `Math.cbrt( )`
4. `Math.log( )`



## V. Distinguish between:

1. Math.ceil( ) and Math.floor( )
2. Math rint( ) and Math.round( )

## VI. Unsolved Programs:

1. Write a program to calculate the value of the given expression:

$$\frac{1}{a^2} + \frac{2}{b^2} + \frac{3}{c^2}$$

Take the values of a, b and c as input from the console. Finally, display the result of the expression to its nearest whole number.

2. For every natural number  $m > 1$ ;  $2m$ ,  $m^2 - 1$  and  $m^2 + 1$  form a Pythagorean triplet. Write a program to input the value of 'm' through console to display a 'Pythagorean Triplet'.

Sample Input: 3

Then  $2m = 6$ ,  $m^2 - 1 = 8$  and  $m^2 + 1 = 10$

Thus 6, 8, 10 form a 'Pythagorean Triplet'.

3. Write a program to input a number. Calculate its square root and cube root. Finally, display the result by rounding it off.

Sample Input: 5

Square root of 5 = 2.2360679

Rounded form = 2

Cube root of 5 = 1.7099759

Rounded form = 2

4. The volume of a sphere is calculated by using formula:

$$v = \frac{4}{3} * \frac{22}{7} * r^3$$

Write a program to calculate the radius of a sphere by taking its volume as an input.

**Hint:** radius =  $\sqrt[3]{\text{volume} * \frac{3}{4} * \frac{7}{22}}$

5. A trigonometrical expression is given as:

$$\frac{\tan A - \tan B}{1 + \tan A * \tan B}$$

Write a program to calculate the value of the given expression by taking the values of angles A and B (in degrees) as input.

**Hint:** radian =  $\left(\frac{22}{7 * 180}\right) * \text{degree}$

6. The standard form of quadratic equation is represented as:

$$ax^2 + bx + c = 0$$

where  $d = b^2 - 4ac$ , known as 'Discriminant' of the equation.

Write a program to input the values of a, b and c. Calculate the value of discriminant and display the output to the nearest whole number.



LEARNING SCOPE

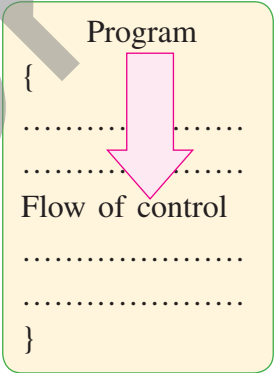
- Normal flow of control
- Conditional flow of control: if, if-else, if-else-if and nested if
- System.exit(0) - to terminate the program
- Multiple branching of control: switch case, fall through in switch case, Menu driven programs.

Computer program is defined as a set of valid statements designed to perform a specific operation. When a command is issued to execute a program, the control reaches the first line of the program. It keeps executing the statements one by one unless the end of the program is reached. The movement of control from one line to other is known as ‘Flow of Control’.

The flow of control is maintained in the following ways:

1. Normal Flow of Control

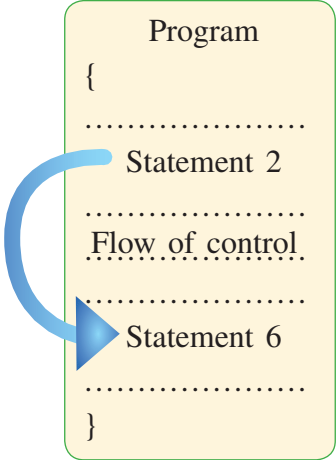
When all the statements of a program is executed one by one from beginning to end (without skipping any statement), the flow of control is known as Normal Flow of Control.



2. Conditional Flow of Control

A set of statements that are grouped together in a program is known as a block. During the course of programming, it may happen that we want to execute a block by ignoring another block based on a given condition. In this situation, the control is diverted to perform the task accordingly. It is known as Conditional Flow of Control. Some of the statements such as if, switch, for, while, etc. are used to direct the flow of control conditionally.

In this unit, you will learn the conditional statements by using ‘if’ construct and multiple branching statement.



I. If constructs

If construct is used to direct the control to execute either of the two blocks of statements based on a given condition. It is available with the following modifications:

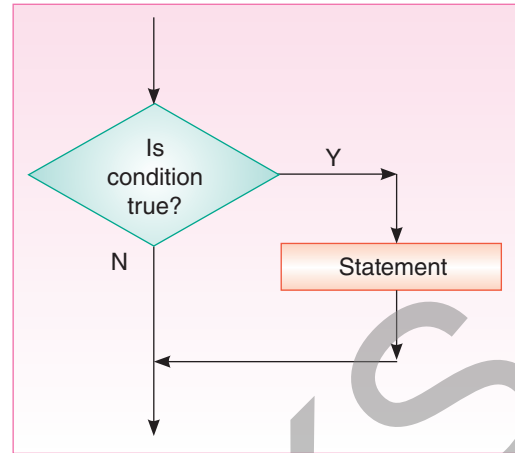


## • if statement

This statement checks the condition first. If the condition is true then the next statement is executed otherwise it is ignored.

```
e.g.,  
if(a>10)  
    c = a*a;  
System.out.println(c);
```

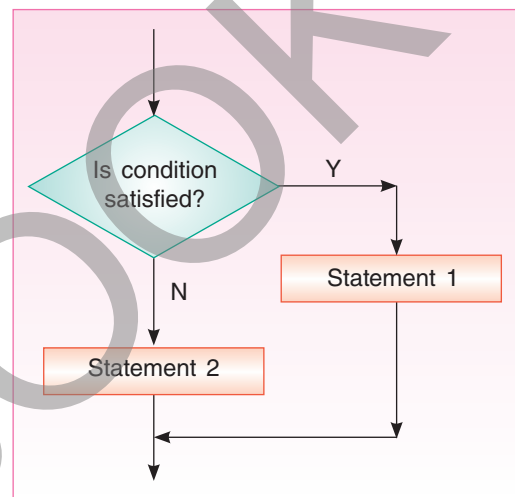
In the example shown above, if the given condition (i.e., a is greater than 10) is true then it will calculate the square of the number 'a' otherwise, it will ignore this statement and will go to the next statement. In that case, it will print the default value of c as 0.



## • if-else statement

If you want to execute either of the two statements depending upon a given condition then 'if-else' statement is used.

```
e.g.,  
if(a>b)  
    max=a;  
else  
    max=b;
```



The example shown above illustrates that max is assigned either with the value 'a' or 'b' depending upon the condition (a>b) is true or false.

If there are multiple statements to be executed when the condition is true or false then they must be enclosed within curly braces { }.

For example,

```
if(a>b)  
{  
    max=a;  
    d=a-b;  
}  
else  
{  
    max=b;  
    d=b-a;  
}
```

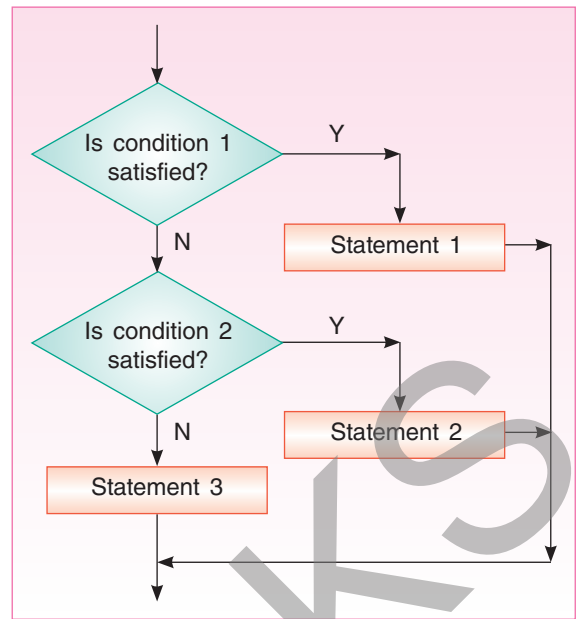


### • if-else-if statement

Sometimes, it may happen that the given condition is false and the user wants to check another condition for taking necessary action, under this situation 'if-else-if' statement is used.

```
e.g.,  
if(marks>=75)  
System.out.println("Distinction");  
elseif(marks>=60)  
System.out.println("First Division");  
elseif(marks>50)  
System.out.println("Second Division");  
else  
System.out.println("Third Division");
```

In this example, the control first of all checks the condition whether  $\text{marks} \geq 75$ . If it is true, it will print "Distinction" and ignore rest of the statements. If the first condition is false, then it will check the next elseif condition whether  $\text{marks} \geq 60$ . If it is true, it will display "First Division" and ignore rest of the statements otherwise, it will check next condition. The process will further be followed accordingly.

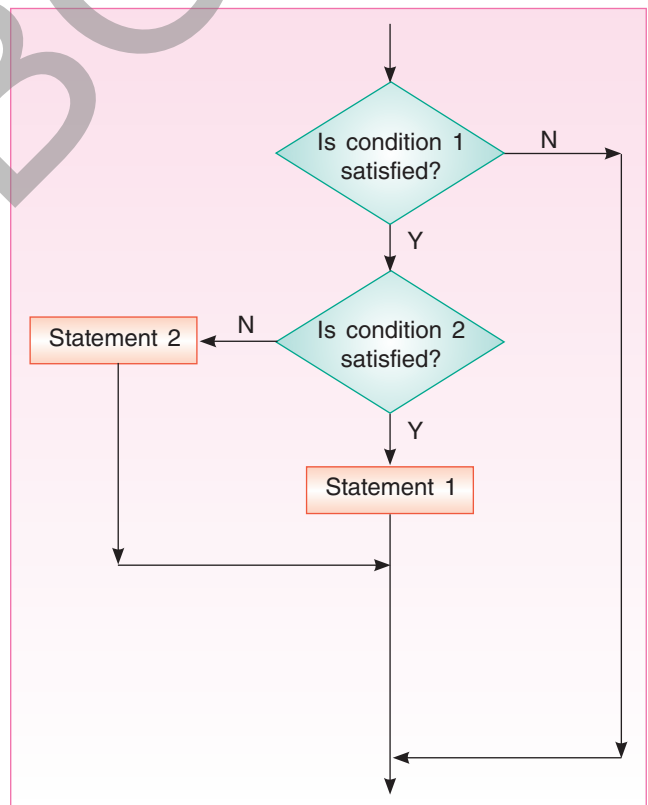


### • Nested if statement

When 'if' statement is used within another 'if' statement, the construct is said to be a 'Nested if' statement. In this construct, if the given condition is true then it further checks another if condition and carries out the instructions accordingly.

```
e.g.,  
if(a>=10)  
{  
    if(a<100)  
        c=a*a;  
}  
else  
    c = a*a*a;
```

Referring to the snippet shown above, first of all the condition ( $a \geq 10$ ) is checked. If it is true then the control enters the block and checks another condition ( $a < 100$ ). If this condition is true, the square of 'a' will be calculated. In case, the first condition is false then the control enters into the else part and calculates the cube of the number.



### Points to Remember!

1. 'if' statement is used to check a condition and when the condition is true, the following statement or a set of statements is executed. In case the condition is false, the control ignores the remaining statements and moves to the next statement of the program.
2. When you need to execute more than one statement(s) based on a condition then all the statements must be enclosed within curly brackets { } inside if block.
3. A statement that includes one or more statements within it marked under curly brackets { } is said to be a compound statement.

## Unusual Termination of a Program (System.exit(0))

Sometimes, you may need to terminate a program even if all of its statements have not been executed. The function *System.exit(0)* is used when you want to terminate the execution at any instance of the program. As soon as '*System.exit(0)*' function is invoked, it terminates the execution ignoring the rest of the statements of the program.

**Syntax:** *System.exit(0);*

e.g.,

```
System.out.println("Enter a number");
n = in.nextInt();
if(n<0)
{
    System.out.println("It is a negative number.");
    System.out.println("The program terminates.");
    System.exit(0);
}
sqrt=Math.sqrt(n);
System.out.println("Square root of " + n + "="+sqrt);
```

In the snippet shown above, the value of 'n' is checked whether it is negative or not. If it is a negative number then program will be terminated by using *System.exit(0)* function after displaying two given messages. In case, the number is positive then only square root of the number will be calculated.

## II. Multi-Branching statement

It is another type of conditional flow of control in which one out of multiple blocks of statements is executed, based on a given condition. Generally, we use 'Switch case' statement to handle multiple branching flow of control.

### Switch Statement

In a switch case statement, a number of blocks are created under different cases. All the cases are enclosed within curly braces { } under switch statement. A particular case is executed based on the given value (user's choice) of control variable passed as an argument to the *switch( )* statement.



```
e.g.,
switch(a)
{
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");
}
```

In the construct shown above, a number of cases are enclosed within switch statement. A particular case will be executed when matched with the value of control variable ‘a’. The default case will be executed only when the value of ‘a’ doesn’t match with any of the cases listed above.

**Terms related to switch statement**

• **Control Variable:**

The variable passed as an argument to the switch statement that decides which case is to be executed is known as a control variable.

• **Break Statement:**

This is the statement used at the end of each case which acts as the case terminator. As soon as break is encountered, the control is forced to move out of the switch block.

• **Default Case:**

If no case is matched in the switch block for a given value of control variable, default case is executed implicitly.

Control Variable

```
switch(n) ←
{
case 1:
-----;
-----;
break;
case 2:
-----;
-----;
break;
default:
statement
}
```

```
switch(n)
{
case 1:
-----;
-----;
case 2:
-----;
-----;
case 3:
-----;
-----;
break;
}
```

Missing  
break  
↓  
(Fall Through)

• **Fall Through:**

In case a break statement is not used at the end of a case, the control enters into the next case for execution. This condition is said to be Fall Through.



## SOLVED PROGRAMS BY USING CONDITIONAL STATEMENTS

**Prog. 1** Write a program to input three different single digit numbers between 1 and 9 (both inclusive). Display the greatest and smallest three digit number.

Sample Input : 2, 7, 4

Sample Output :

Greatest three digit number = 742

Smallest three digit number = 247

// To display the greatest and smallest number formed by digits

```
import java.util.*;
```

```
public class Number
```

```
{
```

```
public static void main(String args[ ])
```

```
{
```

```
Scanner in=new Scanner(System.in);
```

```
int a,b,c,gn=0,sn=0;
```

```
System.out.println("Enter three digits between 1 and 9");
```

```
a=in.nextInt();
```

```
b=in.nextInt();
```

```
c=in.nextInt();
```

```
if((a>b)&&(a>c))
```

```
{
```

```
if(b>c)
```

```
{
```

```
gn=100*a+10*b+c;
```

```
sn=100*c+10*b+a;
```

```
}
```

```
else
```

```
{
```

```
gn=100*a+10*c+b;
```

```
sn=100*b+10*c+a;
```

```
}
```

```
}
```

```
if((b>a)&&(b>c))
```

```
{
```

```
if(a>c)
```

```
{
```

```
gn=100*b+10*a+c;
```

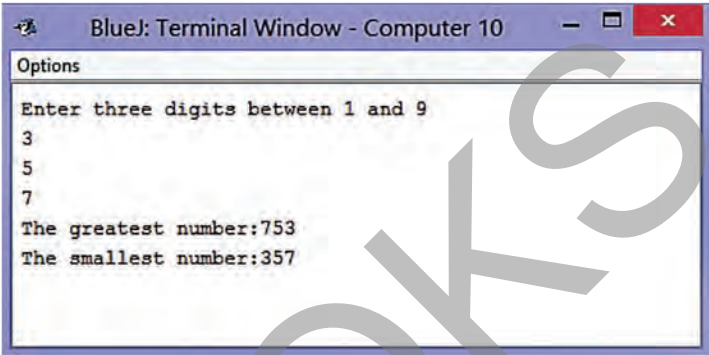
```
sn=100*c+10*a+b;
```

```
}
```



```
else
{
gn=100*b+10*c+a;
sn=100*a+10*c+b;
}
}
if((c>a)&&(c>b))
{
if(a>b)
{
gn=100*c+10*a+b;
sn=100*b+10*a+c;
}
else
{
gn=100*c+10*b+a;
sn=100*a+10*b+c;
}
}
System.out.println("The greatest number:"+gn);
System.out.println("The smallest number:"+sn);
}
}
```

Execution of the program



Prog. 2

The Electricity Board charges from their consumers according to the units consumed per month. The amount is calculated as per the tariff given below.

| Units consumed      | Charges     |
|---------------------|-------------|
| up to 100 units     | ₹ 5.50/unit |
| for next 200 units  | ₹ 6.50/unit |
| for next 300 units  | ₹ 7.50/unit |
| more than 600 units | ₹ 8.50/unit |

Write a program to input consumer’s name, consumer’s number and the units consumed. The program displays the following information after calculating the amount.

Money Receipt

Consumer’s Number: .....  
Consumer’s Name: .....  
Units consumed: .....  
Amount to be paid: .....

```
// To calculate the Electricity Bill
import java.util.*;
public class Electricity_Bill
```

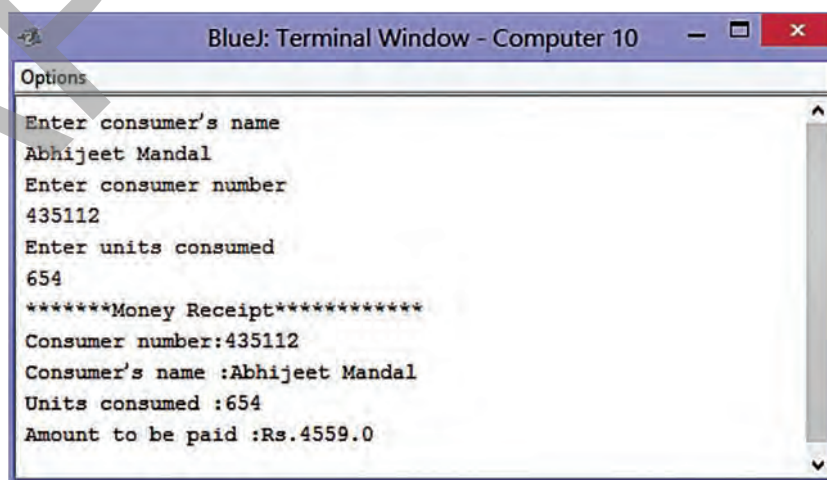


```

{
public static void main(String args[])
{
Scanner in=new Scanner(System.in);
int n,u;
String name;
double amt=0,total=0;
System.out.println("Enter consumer's name");
name=in.nextLine();
System.out.println("Enter consumer's number");
n=in.nextInt();
System.out.println("Enter units consumed");
u=in.nextInt();
if(u<=100)
amt=u*5.50;
if((u>100)&&(u<=300))
amt=100*5.50+(u-100)*6.50;
if((u>300)&&(u<=600))
amt=100*5.50+200*6.50+(u-300)*7.50;
if(u>600)
amt=100*5.50+200*6.50+300*7.50+(u-600)*8.50;
System.out.println("*****Money Receipt*****");
System.out.println("Consumer's number:"+n);
System.out.println("Consumer's name :"+name);
System.out.println("Units consumed :"+u);
System.out.println("Amount to be paid :Rs."+amt);
}
}

```

## Execution of the program



```

BlueJ: Terminal Window - Computer 10
Options
Enter consumer's name
Abhijeet Mandal
Enter consumer number
435112
Enter units consumed
654
*****Money Receipt*****
Consumer number:435112
Consumer's name :Abhijeet Mandal
Units consumed :654
Amount to be paid :Rs.4559.0

```





**Prog. 3**

A cloth showroom has announced the following festival discounts and the assured gifts on the purchase of items, based on the total cost of the items purchased:

| Total Cost                | Discount | Assured Gift  |
|---------------------------|----------|---------------|
| Less than or up to ₹ 2000 | 5%       | Wall Clock    |
| ₹ 2001 to ₹ 5000          | 10%      | School Bag    |
| ₹ 5001 to ₹ 10,000        | 15%      | Electric Iron |
| More than ₹ 10,000        | 20%      | Wrist Watch   |

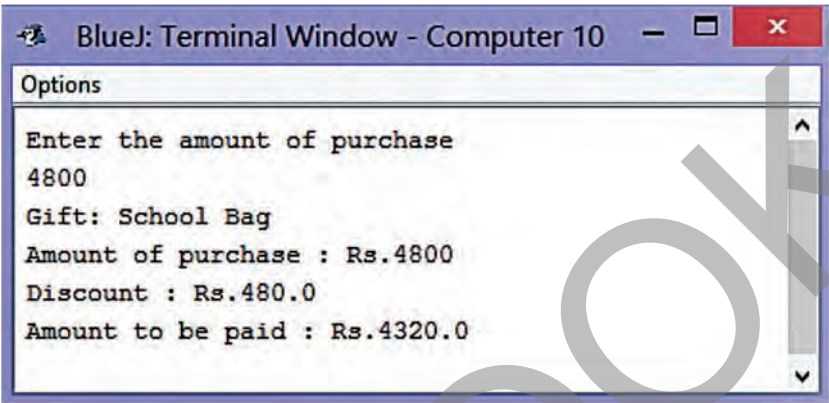
Write a program to input the total cost of the item purchased, discount, amount to be paid after availing discount and the assured gift.

```
// To calculate the discount
import java.util.*;
public class Discount
{
    public static void main(String args[])
    {
        Scanner in=new Scanner(System.in);
        int p;
        double d=0,amt=0;
        System.out.println("Enter the amount of purchase");
        p=in.nextInt();
        if(p<=2000)
        {
            d=p*5.0/100.0;
            System.out.println("Gift: Wall Clock");
        }
        if(p>=2001&&p<=5000)
        {
            d=p*10.0/100.0;
            System.out.println("Gift: School Bag");
        }
        if(p>=5001&&p<=10000)
        {
            d=p*15.0/100.0;
            System.out.println("Gift: Electric Iron");
        }
        if(p>10000)
        {
            d=p*20.0/100.0;
            System.out.println("Gift: Wrist Watch");
        }
    }
}
```



```
}
amt=p-d;
System.out.println("Amount of purchase : Rs."+p);
System.out.println("Discount : Rs."+d);
System.out.println("Amount to be paid : Rs."+amt);
}
}
```

Execution of the program



**Prog. 4**

Given below is a hypothetical table showing rate of income tax for male citizens below the age of 65 years:

| Taxable Income (TI)                                      | Income Tax                                  |
|----------------------------------------------------------|---------------------------------------------|
| does not exceed ₹2,50,000                                | Nil                                         |
| more than ₹2,50,000 and less than or equal to ₹5,00,000  | $(TI - ₹2,50,000) \times 10\%$              |
| more than ₹5,00,000 and less than or equal to ₹10,00,000 | $[(TI - ₹5,00,000) \times 20\%] + ₹34,000$  |
| more than ₹10,00,000                                     | $[(TI - ₹10,00,000) \times 30\%] + ₹94,000$ |

Write a program to input the age, gender (male or female) and taxable income of a person. If the age is more than 65 years or the gender is female, display “Wrong category”. If the age is less than or equal to 65 years and the gender is male, compute and display the income tax payable as per the table given above.

[ICSE 2012]

```
// To calculate and display Income Tax
import java.util.*;
public class Income_Tax
{
public static void main(String args[])
{
int age,ti;
char g;
```

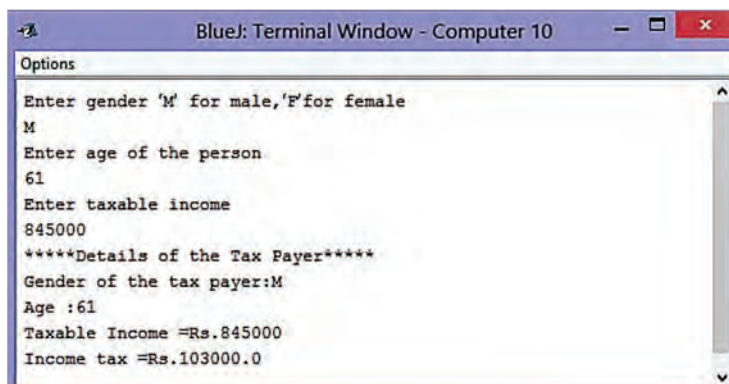


```

double tax=0.0;
Scanner in=new Scanner(System.in);
System.out.println("Enter gender 'M' for male,'F'for female");
g=in.next().charAt(0);
System.out.println("Enter age of the person");
age=in.nextInt();
System.out.println("Enter taxable income");
ti=in.nextInt();
if(age<=65&&g.equals("M"))
{
if(ti<=250000)
tax=0.0;
if(ti>250000 && ti<=500000)
tax=(ti-250000)*10.0/100.0;
if(ti>500000 && ti<=1000000)
tax=((ti-500000.0)*20.0/100.0)+34000.0;
if(ti>1000000)
tax=((ti-1000000)*30.0/100.0)+94000.0;
System.out.println("*****Details of the Tax Payer*****");
System.out.println("Gender of the tax payer:"+g);
System.out.println("Age :"+age);
System.out.println("Taxable Income =Rs."+ti);
System.out.println("Income tax =Rs."+tax);
}
else
{
System.out.println("Wrong category!!");
System.out.println("No calculation of Income Tax!!");
}
}
}

```

### Execution of the program



```

BlueJ: Terminal Window - Computer 10
Options
Enter gender 'M' for male,'F'for female
M
Enter age of the person
61
Enter taxable income
845000
*****Details of the Tax Payer*****
Gender of the tax payer:M
Age :61
Taxable Income =Rs.845000
Income tax =Rs.103000.0

```

**Prog. 5** Write a menu driven program to find the area of an Equilateral triangle, an Isosceles triangle and a Scalene triangle as per the user's choice.

1. Equilateral triangle =  $\frac{\sqrt{3}}{4}s^2$ ,  $s$  = side of an equilateral triangle

2. Isosceles triangle =  $\frac{1}{4}b*\sqrt{4a^2 - b^2}$

3. Scalene triangle =  $\sqrt{s(s-m)(s-n)(s-p)}$ ,  $s = \frac{m+n+p}{2}$   
(where  $m$ ,  $n$  and  $p$  are three sides of a scalene triangle)

// A menu driven program to calculate the area of triangles

```
import java.util.*;
public class Menu
{
public static void main(String args[])
{
Scanner in=new Scanner(System.in);
int c,s,a,b,m,n,p;
double k, area=0;
System.out.println("1. Area of an Equilateral Triangle");
System.out.println("2. Area of an Isosceles Triangle");
System.out.println("3. Area of a Scalene Triangle");
System.out.println("Enter your choice");
c=in.nextInt();
switch(c)
{
case 1:
System.out.println("Enter side of an Equilateral triangle");
s=in.nextInt();
area= (Math.sqrt(3)*s*s)/4.0;
System.out.println("Area="+area);
break;
case 2:
System.out.println("Enter side and base of Isosceles Triangle");
a=in.nextInt();
b=in.nextInt();
area=b*(Math.sqrt(4*a*a-b*b))/4.0;
System.out.println("Area="+area);
break;
case 3:
System.out.println("Enter sides of Scalene triangle");
```



```

m=in.nextInt();
n=in.nextInt();
p=in.nextInt();
k=(m+n+p)/2.0;
area=Math.sqrt(k*(k-m)*(k-n)*(k-p));
System.out.println("Area="+area);
break;
default:
System.out.println("Wrong choice!!");
}
}
}

```

### Execution of the program

```

BlueJ: Terminal Window - Computer 10
Options
1.Area of an Equilateral Triangle
2.Area of an Isosceles Triangle
3.Area of a Scalene Triangle
Enter your choice
3
Enter sides of Scalene triangle
12
10
8
Area=39.68626966596886

```

**Prog. 6** Using the switch statement, write a menu driven program to calculate the maturity amount of a bank deposit. The user is given the following options:

- (i) Term Deposit
- (ii) Recurring Deposit

For option (i) accept principal ( $p$ ), rate of interest ( $r$ ) and time period in years ( $n$ ). Calculate and output the maturity amount ( $A$ ) using the formula:

$$A = p \left( 1 + \frac{r}{100} \right)^n$$

For option (ii) accept monthly instalment ( $p$ ), rate of interest ( $r$ ) and time period in months ( $n$ ). Calculate and output the maturity amount ( $A$ ) using the formula:

$$A = p * n + p * \left( \frac{n * (n + 1)}{2} \right) * \frac{r}{100} * \frac{1}{12}$$

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

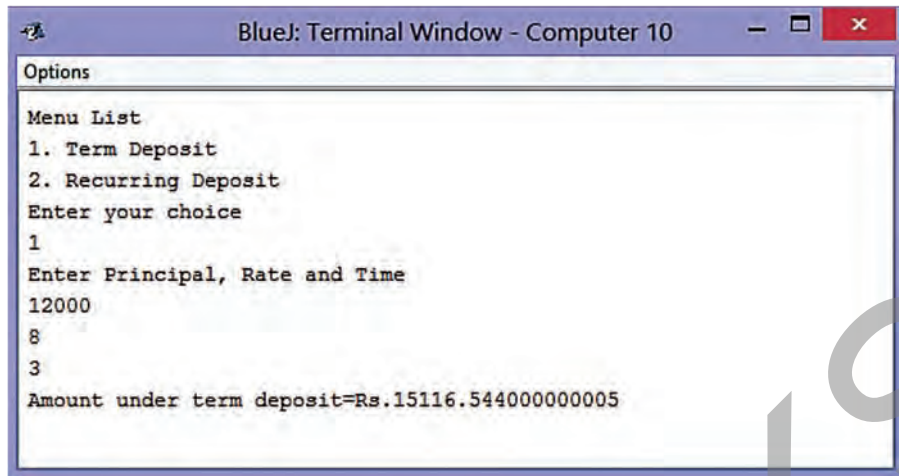
```

// To calculate Term deposit or Recurring deposit
import java.util.*;
public class Deposit
{
public static void main(String args[])
{
Scanner in = new Scanner(System.in);
double p,r,a;
int n,ch;
System.out.println("Menu List");
System.out.println("1. Term Deposit");
System.out.println("2. Recurring Deposit");
System.out.println("Enter your choice");
ch=in.nextInt();
switch(ch)
{
case 1:
System.out.println("Enter Principal, Rate and Time");
p=in.nextDouble();
r=in.nextDouble();
n=in.nextInt();
a=p*Math.pow((1 + r/100.0),n);
System.out.println("Amount under term deposit=Rs."+a);
break;
case 2:
System.out.println("Enter Principal, Rate and Time");
p=in.nextDouble();
r=in.nextDouble();
n=in.nextInt();
a=p*n+p*(n * (n+1))/2.0 * r/100.0 * 1.0/12.0;
System.out.println("Amount under recurring deposit=Rs."+a);
break;
default:
System.out.println("Invalid Choice");
}
}
}

```



## Execution of the program



```
Options
Menu List
1. Term Deposit
2. Recurring Deposit
Enter your choice
1
Enter Principal, Rate and Time
12000
8
3
Amount under term deposit=Rs.15116.544000000005
```

## REVIEW Insight

- Q.1** What is control variable in a switch statement?
- Ans.** In a switch statement, a variable that takes the control to move or jump to a particular case, satisfying the value is called a control variable.  
For example: switch (p)  
Here, p is the control variable.
- Q.2** In a switch statement, why is it necessary to use break after each case?
- Ans.** In a switch statement, the break statement terminates the block. If the break is not used, the control will keep on executing other cases under the switch block.
- Q.3** Use switch case statement for the given program snippet:
- ```
if(a==100)
c=a*2;
if(a==200)
c=a*4;
if(a==600)
c=a*10;
```
- Ans.**
- ```
switch(a)
{
case 100: c=a*2;
break;
case 200: c=a*4;
break;
case 600: c=a*10;
}
```
- Q.4** Use if else statement for the given snippet:
- ```
String st=(a>0) ? (a%2==0) ? "Positive even": "Positive odd": "Negative number";
```
- Ans.**
- ```
String st; int a;
if(a>0)
{
if(a%2==0)
st= "Positive Even";
```



```

else
st= "Positive Odd";
}
else
st= "Negative Number";

```

**Q.5** Find errors in the following snippets:

- (i) 

```
int a=5,b=3;
if(a<=b)
System.out.println(a);
else
System.out.println(c=a+b);
```
- (ii) 

```
int p=10;
switch(p)
{
case1: q=p*2;
break;
case2: q=p+2;
break;
case3: q=p-2;
break;
}
```

- Ans.** (i) Assignment statement can't be used along with the print statement.  
(ii) (a) No execution takes place as the value of p=10.  
(b) The variable q is not declared.

**Q.6** Rewrite the following program segment using if-else statements instead of the ternary operator. [ICSE 2014]

String grade= (marks>= 90)? "A": (marks>=80)? "B": "C";

- Ans.**

```
int marks; String grade;
if(marks>=90)
grade="A";
else
if(marks>=80)
grade="B";
else
grade="C";
```

## EXERCISES

**I. State whether the following statements are 'True' or 'False':**

1. if statement is also called as a conditional statement.
2. A conditional statement is essentially formed by using relational operators.
3. An if-else construct accomplishes 'fall through'.
4. In a switch case, when the switch value does not match with any case then the execution transfers to the default case.
5. The break statement mayn't be used in a switch statement.



## II. Tick the correct answer:

- If  $m, n, p$  are the three integers, then which of the following holds true, if  $(m==n) \&\& (n!=p)$ ?  
(a) 'm' and 'n' are equal (b) 'n' and 'p' are equal  
(c) 'm' and 'p' are equal (d) none
- A compound statement can be stated as:  
(a)  $p = \text{in.nextInt}();$   
     $q = \text{in.nextInt}();$   
(b)  $m = ++a;$   
     $n = --b;$   
(c)  $\text{if}(a > b)$   
     $\{a++; b--;\}$   
(d) none
- If  $((p > q) \&\& (q > r))$  then  
(a)  $q$  is the smallest number (b)  $q$  is the greatest number  
(c)  $p$  is the greatest number (d) none  
(where  $p, q$  and  $r$  are three integer numbers)
- if  $(a < b)$   
     $c = a;$   
    else  
     $c = b;$   
It can be written as:  
(a)  $c = (b < a) ? a : b;$  (b)  $c = (a != b) ? a : b;$   
(c)  $c = (a < b) ? b : a;$  (d) none
- If  $(a < b \&\& a < c)$   
(a)  $a$  is the greatest number (b)  $a$  is the smallest number  
(c)  $b$  is the greatest number (d) none  
(where  $a, b$  and  $c$  are three integer numbers)

## III. Write the Java expressions for the following:

- $\sqrt[3]{ab + cd}$
- $p^3 + q^4 - \frac{1}{2}r$
- $\frac{-b + \sqrt{b^2 - 4ac}}{2a}$
- $\frac{0.05 - 2y^3}{(x - y)}$  [ICSE 2007]
- $\sqrt{mn} + \sqrt[3]{(m + n)}$
- $\frac{3}{4}(a + b) - \frac{2}{5}ab$
- $\frac{3}{8}\sqrt{(b^2 + c^3)}$
- $\sqrt[3]{a} + b^2 - \sqrt[3]{c}$
- $\frac{\sqrt{a + b^2 + c^3}}{3}$
- $\frac{\sqrt{3x + x^2}}{(a + b)}$  [ICSE 2018]

## IV. Predict the output:

- ```
int m=3,n=5,p=4;
if(m==n)&&(n!=p)
{
    System.out.println(m*n);
    System.out.println(n%p);
}
if((m!=n) || (n==p))
{
    System.out.println(m+n);
    System.out.println(m-n);
}
```
- (i)  $p = 1$  (ii)  $p = 3$   

```
int a=1,b=2,c=3;
switch(p)
{
    case 1: a++;
    case 2: ++b;
    break;
    case 3: c--;
}
System.out.println(a + "," + b + "," + c);
```

## V. Convert the following constructs as directed:

1. *switch case* construct into *if-else-if* :

```
switch(n)
{
    case 1:
        s=a+b;
        System.out.println("Sum="+s);
        break;
    case 2:
        d=a-b;
        System.out.println("Difference="+d);
        break;
    case 3:
        p=a*b;
        System.out.println("Product="+p);
        break;
    default:
        System.out.println("Wrong Choice!");
}
```

2. *if-else-if* construct into *switch case*:

```
if(var==1)
    System.out.println("Distinction");
else if(var==2)
    System.out.println("First Division");
else if(var==3)
    System.out.println("Second Division");
else
    System.out.println("invalid");
```

[ICSE 2018]

## VI. Answer the following questions:

1. What is meant by 'conditional' statement? Explain.
2. What is the significance of `System.exit(0)`?
3. Is it necessary to include 'default' case in a switch statement? Justify.
4. What will happen if 'break' statement is not used in a switch case? Explain.
5. When does 'Fall through' occur in a switch statement? Explain.
6. What is a compound statement? Give an example.
7. Explain if-else-if construct with an example.
8. Give two differences between the switch statement and the if-else statement.

[ICSE 2014]

## VII. Unsolved Java Programs:

1. Write a program to input three numbers (positive or negative). If they are unequal then display the greatest number otherwise, display they are equal. The program also displays whether the numbers entered by the user are 'All positive', 'All negative' or 'Mixed numbers'.

Sample Input: 56, -15, 12

Sample Output: The greatest number is 56

Entered numbers are mixed numbers.



2. A triangle is said to be an 'Equable Triangle', if the area of the triangle is equal to its perimeter. Write a program to enter three sides of a triangle. Check and print whether the triangle is equable or not.

For example, a right angled triangle with sides 5, 12 and 13 has its area and perimeter both equal to 30.

3. A special two-digit number is such that when the sum of its digits is added to the product of its digits, the result is equal to the original two-digit number.

Example: Consider the number 59.

Sum of digits = 5 + 9 = 14

Product of digits = 5 \* 9 = 45

Total of the sum of digits and product of digits = 14 + 45 = 59

Write a program to accept a two-digit number. Add the sum of its digits to the product of its digits. If the value is equal to the number input, display the message "Special 2 – digit number" otherwise, display the message "Not a special two-digit number".

[ICSE 2014]

4. The standard form of quadratic equation is given by:

$ax^2 + bx + c = 0$ , where  $d = b^2 - 4ac$ , is known as discriminant that determines the nature of the roots of the equation as:

Condition	Nature
if $d \geq 0$	Roots are real
if $d < 0$	Roots are imaginary

Write a program to determine the nature and the roots of a quadratic equation, taking a, b, c as input. If  $d = b^2 - 4ac$  is greater than or equal to zero, then display 'Roots are real', otherwise display 'Roots are imaginary'.

The roots are determined by the formula as:

$$r1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, r2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

5. An air-conditioned bus charges fare from the passengers based on the distance travelled as per the tariff given below:

Distance Travelled	Fare
Up to 10 km	Fixed charge ₹ 80
11 km to 20 km	₹ 6/km
21 km to 30 km	₹ 5/km
31 km and above	₹ 4/km

Design a program to input distance travelled by the passenger. Calculate and display the fare to be paid.

6. An ICSE school displays a notice on the school notice board regarding admission in Class XI for choosing stream according to marks obtained in English, Maths and Science in Class 10 Council Examination.

Marks obtained in different subjects	Stream
Eng, Maths and Science $\geq 80\%$	Pure Science
Eng and Science $\geq 80\%$ , Maths $\geq 60\%$	Bio. Science
Eng, Maths and Science $\geq 60\%$	Commerce

Print the appropriate stream allotted to a candidate. Write a program to accept marks in English, Maths and Science from the console.



7. A bank announces new rates for Term Deposit Schemes for their customers and Senior Citizens as given below:

Term	Rate of Interest (General)	Rate of Interest (Senior Citizen)
Up to 1 year	7.5%	8.0%
Up to 2 years	8.5%	9.0%
Up to 3 years	9.5%	10.0%
More than 3 years	10.0%	11.0%

The ‘senior citizen’ rates are applicable to the customers whose age is 60 years or more. Write a program to accept the sum (p) in term deposit scheme, age of the customer and the term. The program displays the information in the following format:

Amount Deposited	Term	Age	Interest earned	Amount Paid
xxx	xxx	xxx	xxx	xxx

8. A courier company charges differently for ‘Ordinary’ booking and ‘Express’ booking based on the weight of the parcel as per the tariff given below:

Weight of parcel	Ordinary booking	Express booking
Up to 100 gm	₹ 80	₹ 100
101 to 500 gm	₹ 150	₹ 200
501 gm to 1000 gm	₹ 210	₹ 250
More than 1000 gm	₹ 250	₹ 300

Write a program to input weight of a parcel and type of booking (‘O’ for ordinary and ‘E’ for express). Calculate and print the charges accordingly.

### Menu Driven/Switch Case programs

9. Write a menu driven program to calculate:
- (a) Area of a circle =  $p \cdot r^2$ , where  $p = \frac{22}{7}$
  - (b) Area of a square = side\*side
  - (c) Area of a rectangle = length\*breadth
- Enter ‘c’ to calculate area of circle, ‘s’ to calculate area of square and ‘r’ to calculate area of rectangle.
10. Write a program using switch case to find the volume of a cube, a sphere and a cuboid. For an incorrect choice, an appropriate error message should be displayed.
- (a) Volume of a cube =  $s \cdot s \cdot s$
  - (b) Volume of a sphere =  $\frac{4}{3} \pi \cdot r \cdot r \cdot r$  ( $\pi = \frac{22}{7}$ )
  - (c) Volume of a cuboid =  $l \cdot b \cdot h$
11. The relative velocity of two trains travelling in opposite directions is calculated by adding their velocities. In case, the trains are travelling in the same direction, the relative velocity is the difference between their velocities. Write a program to input the velocities and length of the trains. Write a menu driven program to calculate the relative velocities and the time taken to cross each other.



12. In order to purchase an old car, the depreciated value can be calculated as per the tariff given below:

No. of years used	Rate of depreciation
1	10%
2	20%
3	30%
4	50%
Above 4 years	60%

Write a menu driven program to input showroom price and the number of years the car is used ('1' for one year old, '2' for two years old and so on). Calculate the depreciated value. Display the original price of the car, depreciated value and the amount to be paid.

## NOTES

APC BOON

## Unit VIII: Iterative Constructs in Java

### LEARNING SCOPE

- Types of looping statements: entry controlled loops (for and while loop), exit controlled loop (do-while)
- Break and continue
- Inter conversion from for-while-do while
- Finite and infinite loop, delay loop
- Simple programs illustrating all three loops.

You have already learnt about conditional construct in Java in which a particular block of statements is executed based on a given condition.

In this unit we are going to focus on a construct in which a block of statements gets executed repeatedly unless the required task has been completed. Such repetitive construct is known as an iterative construct or a **Loop**.

*Iteration means repeated execution of a set of statements. This can be achieved by using a loop.*

Based on the flow of control, the looping constructs can be categorised into two types. They are:

- Entry Controlled Loop
- Exit Control Loop

Let us have a brief discussion about these loops.

### Entry Controlled Loop

A looping construct in which the condition is checked in the beginning. If the condition is true, the control is allowed to enter into the loop otherwise the entry will be denied. There are two types of looping construct under this category:

- for Loop:**

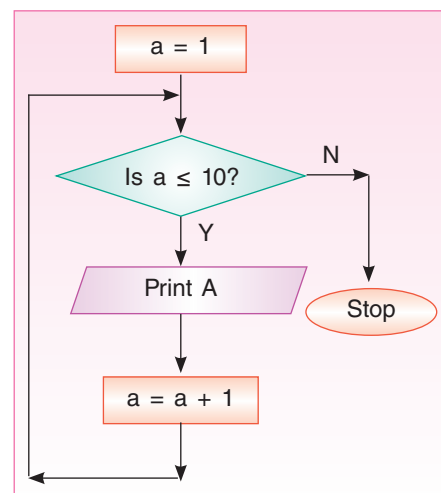
The 'for' loop is used when number of iterations is fixed and known. It is a fixed or known iterative looping construct.

Components of 'for' loop structure:

for (a = 1;	a<=10;	a++)
↓	↓	↓
control variable with initial value	test condition	increment/decrement/step value

e.g.,

```
for(a = 1; a <= 10; a++)  
{  
    System.out.println(a);  
    s = s + a;  
}
```





The given example illustrates that the execution of the loop block will be repeated by using the value of 'a' from 1 to 10 and changing it by increasing the value by 1 after each iteration.

### Important to know

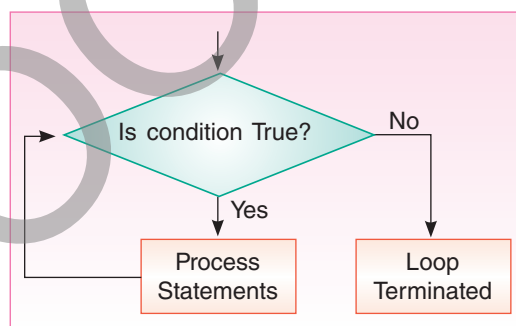
1. If single statement is to be repeated within a loop then it needs not be enclosed within curly braces { }.
2. When a loop runs, the values are updated accordingly as:  
a++ : increases the value by 1  
a-- : decreases the value by 1  
a=a+2 : increases the value by 2  
a=a-2 : decreases the value by 2 and so on.

- **while Loop:**

Sometimes, it may happen that a set of statements are to be executed repeatedly without knowing the number of iterations, while loop is used to solve the purpose. In this looping construct, a block of statements gets executed repeatedly unless the given condition is false. While loop is unfixed or unknown iterative loop.

e.g.,

```
int p=1000,r=10;
while(p<=10000)
{
    si = (p * r) /100.0;
    p = p + si;
}
```



Refer the example shown above. The principle (p) is updated after each iteration in the loop till the given condition is true. As soon as the condition becomes false, the loop will be terminated. Here, the user may not be aware that after how many iterations the value of p will cross 10000. Hence, it is unfixed iterative loop.

## Exit Controlled Loop

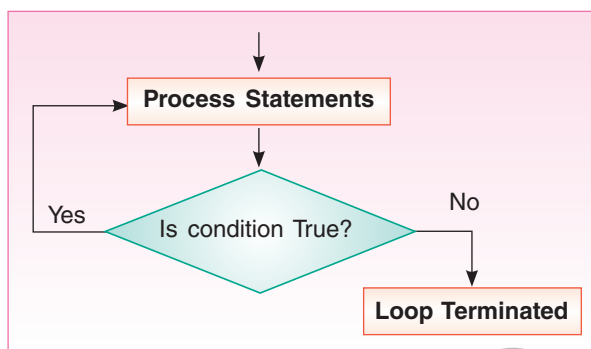
This is another type of looping construct in which condition is checked at the exit point (end point) of the loop. The control will repeat for the next iteration or exit from the loop depending on the condition is true or false respectively.

- **do – while Loop:**

The 'do – while' loop also works in the same way as while loop. It is also unfixed or unknown iterative loop i.e., user may not be aware how many times the iteration will take place. Only difference is the position at which the condition will be checked. In while loop the condition is checked in the beginning of the loop whereas, in do-while loop, the condition is checked at the end of the loop.

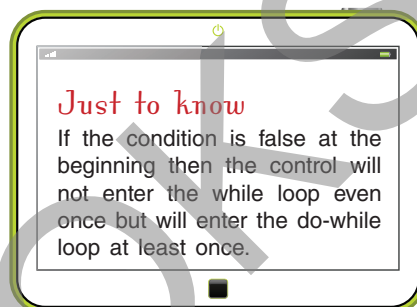
e.g.,  

```
int p = 1000, r = 10;
double si;
do
{
    si = (p * r) / 100.0;
    p = p + si;
}
while(p <= 10000);
```



In the example shown above, the loop will repeat till the given condition is true. You can either use while or do-while loop as per the requirement.

Known and unknown iterative constructs are interchangeable. It depends on the user to decide which loop is suitable in the logic development.



## Break statement used in Loops

Sometimes, it may happen that the loop has not been completed but the requirement has been fulfilled. The user then would like to terminate the loop without its completion. In this situation, ‘break’ statement is used to solve the purpose. Hence, ‘break’ statement is used in a loop to send the control out of the loop, if certain condition is true.

```
for(i=1; i<=10; i++)
{
    if(i == 5)
    break;
    s = s + i;
}
```

for loop

```
i = 1;
while(i<=10);
{
    if(i == 5)
    break;
    s = s + i;
    i++;
}
```

while loop

```
i = 1;
do
{
    if(i == 5)
    break;
    s = s + i;
    i++;
}
while(i<=10);
```

do-while loop

All three snippets shown above are written by using different looping constructs for the same operation. Each loop will add the numbers from 1 to 4. As soon as the value of ‘i’ becomes 5, the break statement will be executed. As a result control will exit from the loop.

## Continue statement used in Loops

The ‘continue’ statement is opposite to the break statement. When continue statement is invoked, the control goes back to check the condition of the loop by ignoring rest statements of the loop.



```

i = 1;
for(i=1; i<=10; i++)
{
    if(i == 5)
        continue;
    s = s + i;
}

```

for loop

```

i = 1;
while(i<=10)
{
    s = s + i;
    if(i == 5)
        continue;
    i++;
}

```

while loop

```

i = 1;
do
{
    s = s + i;
    if(i == 5)
        continue;
    i++;
}
while(i<=10);

```

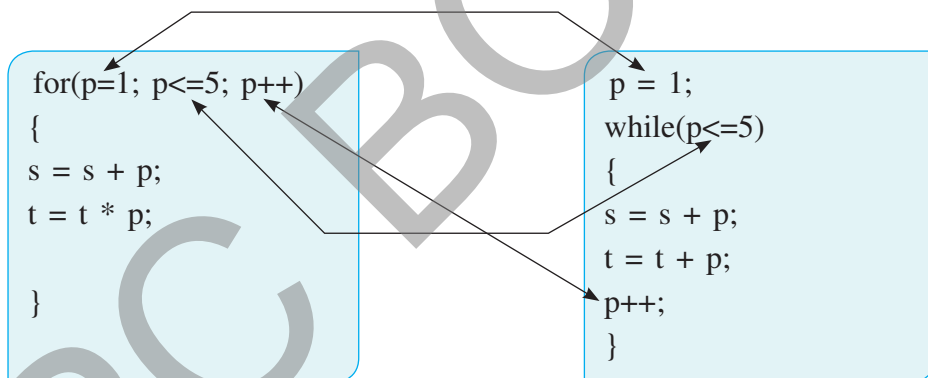
do-while loop

Refer the snippets shown above. They are designed to add the values from 1 to 10 except 5. As soon as, the value of 'i' becomes 5, the continue statement will be executed. As a result, the control will be forced to move back for the next iteration by ignoring the last statement of the loops.

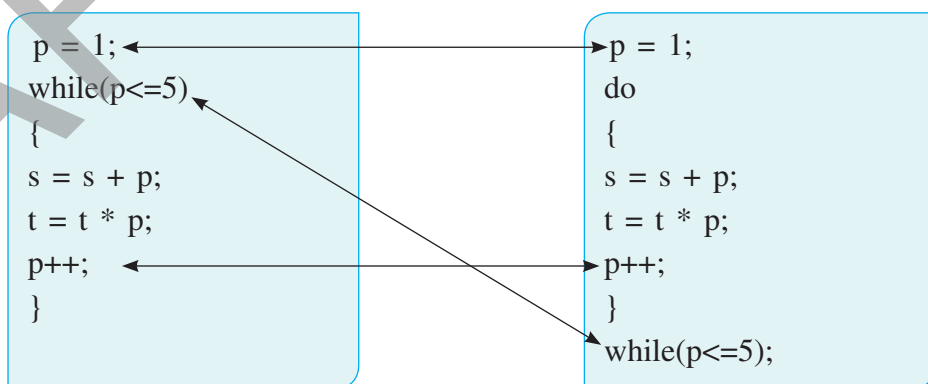
## Interconversion of the Loops

A loop can be converted from one form to other. The system of conversion is as shown below:

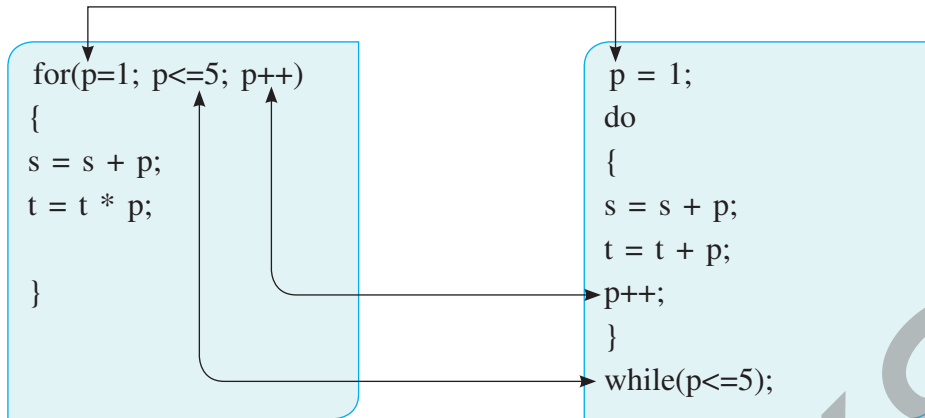
- for loop to while loop



- while loop to do-while loop



- for loop to do-while loop



### User Controlled Loop

Sometimes, you may need to manage a loop which makes iterations based on user's needs. Such a looping structure is termed as a '*User Controlled*' loop.

*For example,*

```

int i=4,j=1,p;
String m= "Yes";
while(m.equals("Yes"))
{
    p=i*j;
    System.out.println(p);
    j++;
    System.out.println("Want to continue...(Yes/No)");
    m=in.readLine( );
}

```

Here, you will find that the loop will generate the multiplication table of 4 (four). However, after each iteration, the program will ask the confirmation from the user whether want to continue or not.

If the user enters "Yes" then the program continues for the next iteration. The program terminates as soon as the user enters his choice as "No".

### Some important terms used with loops

A loop can be used with different modifications to meet our requirements. Based on the way of their functioning, loops can be categorised in the following forms:

- **Finite Loop:** A loop that has natural termination is known as finite loop. It can be created by using all type of looping constructs.

e.g.,



```
for(i=1;i<=5;i++)
{
-----;
-----;
-----;
}
```

for loop

```
a = 1
while(a<=10)
{
-----;
-----;
a++;
}
```

while loop

The looping constructs shown above have definite end. Hence, they are finite iterative loops.

- **Infinite Loop:** A loop that repeats for infinite number of iterations is known as infinite loop. It is also called endless loop.

e.g.

```
for( ; ; )
{
-----;
-----;
-----;
}
```

for loop

```
while(true)
{
-----;
-----;
-----;
}
```

while loop

In the left snippet shown above, all the parameters are missing in 'for' loop. Hence, it will repeat for infinite number of times. In the snippet shown right side, while loop has parameter 'true'. This will also create an endless number of iterations.

- **Empty Loop or Null Loop:** The loop in which body part is missing (i.e., no statement is associated in its body) is called empty loop or null loop. In other words it is a loop that contains null statements.

e.g.

```
for(i=1;i<=10;i++);
or
for(i=1;i<=10;i++)
{ }
```

for loop

```
i = 1;
while(i<=10)
{ }
```

while loop

Refer the snippets shown above. On the left side, for loop is terminated with a semi column. This shows that no statement will be executed within the loop. In this situation, the control will simply repeat 10 times on the same position. The other for statement (in the same snippet) contains empty curly braces {}. This also shows that the loop will repeat 10 times without executing any statement.

In the snippet shown right side, while loop contains empty paranthesis. This also illustrates empty loop.

The empty loops do not perform any meaningful operation but are used in a program to create delay in execution. Hence, it is also referred to as a **delay loop**.

Sometimes, we need to display the values one after the other at a gap of defined time (say, 1 sec), delay loop will be helpful in this case.

## SOLVED PROGRAMS ON ITERATION THROUGH LOOPS

### Prog. 1

Write a program to input a number. Check and display whether it is a Niven number or not.

(A number is said to be Niven if it is divisible by the sum of its digits).

Sample Input : 126

Sum of its digits =  $1 + 2 + 6 = 9$  and 126 is divisible by 9 [ICSE-2016]

```
// To check Niven number
import java.util.*;
class Niven
{
public static void main(String args[])
{
Scanner in = new Scanner(System.in);
int num,sum=0,d,p;
System.out.println("Enter a number");
num = in.nextInt();
p=num;
do
{
d=num%10;
sum=sum+d;
num=num/10;
}
while(num!=0);
if(p%sum==0)
System.out.println("Niven Number");
else
System.out.println("Not a Niven Number");
}
}
```



**Prog. 2**

Write a program to accept a number and check whether the number is perfect or not. A number is said to be perfect if the sum of the factors (including 1 and excluding the number itself) is the same as the original number.

Sample Input : 6

Sample Output : It is a perfect number.

The factors of 6 = 1, 2, 3 and  $1 + 2 + 3 = 6$

```
// To check whether a number is perfect or not
import java.util.*;
public class Perfect
{
    public static void main (String args[])
    {
        Scanner in=new Scanner(System.in);
        int a,n,s=0;
        System.out.println("Enter a number");
        n=in.nextInt();
        for(a=1;a<n;a++)
        {
            if(n%a==0)
            s=s+a;
        }
        if(s==n)
        System.out.println( a + " is a perfect number");
        else
        System.out.println( a + " is not a perfect number");
    }
}
```

**Prog. 3**

Write a program to enter a number and check whether the number is 'Neon' or not. A number is said to be 'Neon' if sum of the digits of the square of the number is equal to the number itself.

Sample Input : 9

Sample Output :  $9*9 = 81$ ,  $8 + 1 = 9$   
: 9 is a Neon number.

```
// To check whether a number is Neon or not
public class Neon
{
    public static void main(int n) // Enter a number
    {
        int p,s=0,d;
        p=n*n;
```





```

do
{
d= p%10;
s=s+d;
p=p/10;
}
while(p!=0);
if(s==n)
System.out.println("Neon number");
else
System.out.println("Not a Neon number");
}
}

```

**Prog. 4** Write a program in Java to enter a number and check whether the number is an Armstrong number or not.  
 (A number is said to be Armstrong if the sum of the cubes of its digits is equal to the original number.)  
 Sample Input : 153  
 Sample Output : 153 is an Armstrong Number because  $1^3 + 5^3 + 3^3 = 153$

```

// To check whether a number is Armstrong or not
public class Armstrong
{
public static void main(int n)// Enter a number
{
int a,num,s=0;
num=n;
while(n>0)
{
a=n%10;
s=s+a*a*a;
n=n/10;
}
if(num==s)
System.out.println("The number" +num + "is an Armstrong Number");
else
System.out.println("The number" +num + "is not an Armstrong Number");
}
}

```



**Prog. 5**

An ‘Automorphic’ number is the number which is contained in the last digit(s) of its square. Write a program to input a number and check whether the number is an ‘Automorphic’ or not.

Sample Input : 25

Solution : The square of 25 is 625 and 25 is present as the last two digits.

Sample Output : 25 is an Automorphic Number.

```
// To check Automorphic number
import java.util.*;
class Automorphic
{
public static void main(int num)
{
int p,f=0;
p=num*num;
do
{
if(num%10!=p%10)
{
f=1;
break;
}
else
{
num=num/10;
p=p/10;
}
}
while(num>0);
if(f==0)
System.out.println("Automorphic Number");
else
System.out.println("Not an Automorphic Number");
}
}
```

**Prog. 6**

Write a program to accept two numbers and find the Greatest Common Divisor (G.C.D) of those numbers.

Sample Input: 25, 45

Sample Output: The Greatest Divisor: 5

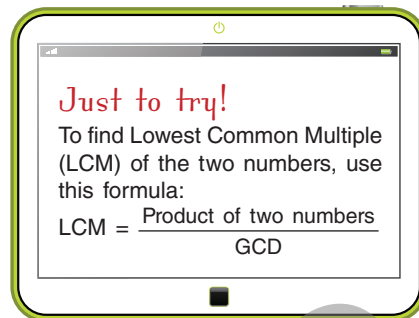
```
//To calculate the GCD of two numbers
```



```

import java.util.*;
public class GCD
{
public static void main (String args[])
{
Scanner in=new Scanner(System.in);
int a,b,i,p,gcd=0;
System.out.println("Enter two numbers");
a=in.nextInt();
b=in.nextInt();
p=a*b;
for(i=1;i<p;i++)
{
if(a%i==0 && b%i==0)
gcd=i;
}
System.out.println("The GCD of two numbers: "+gcd);
}
}

```



#### Prog. 7

You want to calculate the sum of all positive even numbers and sum of all negative odd numbers from a set of numbers. You can enter 0 (zero) to quit the program and thus it displays the result. Write a program to perform the above task.

// To find the sum of all positive even and negative odd numbers as per user's choice

```

import java.util.*;
public class Choice
{
public static void main (String args[])
{
Scanner in=new Scanner(System.in);
int n,s1=0,s2=0;
System.out.println("Enter a number or zero to quit");
n=in.nextInt();
while(n!=0)
{
if(n>0 && n%2==0)
s1=s1+n;
if(n<0 && n%2!=0)
s2=s2+n;
System.out.println("Do you want to continue?");
System.out.println("Enter a number to continue or zero to quit");
}
}
}

```



```

n=in.nextInt();
}
System.out.println("The sum of positive even numbers = "+s1);
System.out.println("The sum of negative odd numbers = "+s2);
}
}

```

**Prog. 8** In the game of tossing a coin, you want to know the number of times you get 'Head' and 'Tail' respectively. You keep the record as '1' (one) for getting 'Head' and '0' (zero) for 'Tail'. Write a program to perform the above task. Suppose you have tossed a coin for 20 times in this game.

```

// To find the frequency of Head and Tail
import java.util.*;
public class Toss
{
public static void main (String args[])
{
Scanner in=new Scanner(System.in);
int i,c,h=0,t=0;
double d;
for(i=1;i<=20;i++)
{
d=Math.random()*2;
c=(int) d;
if(c==1)
h=h+1;
else
t=t+1;
}
System.out.println("Number of times Head obtained = "+h);
System.out.println("Number of times Tail obtained = "+t);
}
}

```

**Prog. 9** Write a program to find the sum of the following series:  

$$S = \frac{1}{2} - \frac{2}{3} + \frac{3}{4} - \frac{4}{5} + \dots \text{to } n$$

```

// To find the sum of the series
import java.util.*;
public class Series
{
public static void main (String args[])

```



```

{
Scanner in=new Scanner(System.in);
int a,n;
double s=0;
System.out.println("Enter the value of n");
n=in.nextInt();
for(a=1;a<=n;a++)
{
if(a%2==0)
s=s-(double)a/(a+1);
else
s=s+(double)a/(a+1);
}
System.out.println("The sum of the series = "+s);
}
}

```

**Prog. 10** Write a program to find the sum of the series given below, taking the values of 'a' and 'n' from the user.

$$S = \frac{a}{2} + \frac{a}{3} + \frac{a}{4} + \dots \text{to } n.$$

// To find the sum of series by using Scanner Class

```

import java.util.*;
public class Series
{
public static void main(String args[])
{
Scanner in = new Scanner(System.in);
System.out.println("Enter the value of a and n");
int a,i,n;
double s=0;
a= in.nextInt();
n= in.nextInt();
for(i=1;i<n;i++)
s=s+(double)a/(i+1);
System.out.println("Sum of the series =" +s);
}
}

```



**Prog. 11**

Shashi Travels Pvt. Ltd. gives the following discount to its customers as per the given tariff:

Ticket Amount	Discount
above ₹ 70,000	18%
₹ 55,001 to ₹ 70,000	16%
₹ 35,001 to ₹ 55,000	12%
₹ 25,001 to ₹ 35,000	10%
Less than ₹ 25,001	2%

Write a program to input the name and the amount of ticket for the customer. Calculate the discount and net amount to be paid. Display the output in the following format for each customer:

Sl. No	Name	Ticket Charge	Discount	Net Amount
1	XXXXX	XXXXXX	XXXXX	XXXXXX

(Assume that there are 15 customers, first customer is given the serial number 1, the next customer 2, and so on). [ICSE 2010]

```
// To calculate discount on ticket
import java.util.*;
public class Ticket
{
    public static void main (String args[])
    {
        Scanner in=new Scanner(System.in);
        {
            int i,p;
            double d=0,amt=0;
            String name;
            for(i=1;i<=15;i++)
            {
                System.out.println("Enter name of the customer :");
                name=in.nextLine();
                System.out.println("Enter the amount of the ticket:");
                p=in.nextInt();
                if(p<25001)
                    d= p*2.0/100.0;
                if(p>25001&&p<=35000)
                    d=p*10.0/100.0;
                if(p>35001&&p<=55000)
                    d=p*12.0/100.0;
                if(p>55001&&p<=70000)
                    d=p*16.0/100.0;
                if(p>70000)
```

```

d=p*18.0/100.0;
amt= (p-d);
System.out.println("Sl.No.\tName\t\tTicket Charge\t\tDiscount\tNet Amount");
System.out.println(i+"\t"+name+"\t\t"+p+"\t\t"+d+"\t\t"+amt);
}
}
}
}

```

**Prog. 12** Using switch statement, write a menu driven program:

- (i) To check and display whether a number input by the user is a composite number or not. (A number is said to be composite if it has one or more factors excluding one and the number itself).  
Example: 4, 6, 8, 9 ...
- (ii) To find the smallest digit of an integer that is input.  
Sample input: 6524  
Sample output: Smallest digit is 2.  
For an incorrect choice, an appropriate error message should be displayed. [ICSE 2013]

/\* A menu driven program to display a composite number or find the smallest digit of an integer\*/

```

import java.util.*;
class Number
{
public static void main(String args[])
{
Scanner in = new Scanner(System.in);
int n,ch,c,i,min,r;
System.out.println("1. To check a composite number");
System.out.println("2. To find the smallest digit of an integer");
System.out.println("Enter your choice!");
ch = in.nextInt();
switch(ch)
{
case 1:
System.out.println("Enter a number to check whether it is a composite or not");
n=in.nextInt();
c=0;
for(i=2; i<n;i++)
if(n%i==0)
c=c+1;
if(c>= 1)

```





```
System.out.println("It is a composite number.");
else
System.out.println("It is not a composite number.");
break;
case 2:
System.out.println("Enter a number to find the smallest digit");
n=in.nextInt();
min=n%10;
while(n>0)
{
r=n%10;
if(r<min)
min=r;
n=n/10;
}
System.out.println("The smallest digit is:\t"+min);
break;
default:
System.out.println("Invalid Choice!");
}
}
}
```



## REVIEW Insight

**Q.1** Differentiate between a Null loop and an Infinite loop.

**Ans.** A null loop doesn't contain any statement in its body whereas an infinite loop repeats the execution for an endless iterations.

**Q.2** What is the similarity between while and do-while loop?

**Ans.** While and do-while repeat the execution of the statements till the given condition is true.

**Q.3** What is meant by an endless loop? Give an example.

**Ans.** An endless loop repeats the execution of statements to an infinite number of times. In this looping process, the control doesn't get terminated on its own.

e.g.,

```
for( ; ; )
{
    body of the loop
}
```

**Q.4** Give an example to illustrate null loop.

**Ans.** The example of null loop by using for statement is as shown below-

```
for(j=1; j<=10; j++)
{ }
```

There is no statement enclosed under curly braces. Hence, the loop simply repeats 10 times.

**Q.5** What is a delay loop?

Or

Why is null loop called delay loop?

**Ans.** A null loop is also called delay loop which does not repeat the execution of any statement but keeps the control engaged for sometimes until the iterations are completed.

**Q.6** Find the errors in the following program snippet:-

```
for(int k=1;k<=10;k--)
    System.out.println(k);
```

**Ans.** In the loop, the update expression should be k++.

**Q.7** Rewrite the program by using while loop:

```
int a,b=0,c;
System.out.println("The numbers are :");
for(a=1;a<=10;a++)
{
    c=b*b;
    System.out.print(c);
    b=b+1;
}
```

**Ans.**

```
int a=1,b=0,c;
System.out.println("The numbers are :");
while(a<=10)
{
    c=b*b;
    System.out.print(c);
    a=a+1;
    b=b+1;
}
```



**Q.8** Convert the following while loop to the corresponding for loop.

[ICSE 2016]

```
int m =5, n =10;
while(n>=1)
{
    System.out.println(m*n);
    n - -;
}
```

**Ans.**

```
int n, m=5;
for(n=10;n>=1;n- -)
    System.out.println(m*n);
```

**Q.9** Analyse the given program segment and answer the following questions.

```
for (int m=5;m<=20;m+=5)
{ if (m%3== 0)
    break;
  else
    if (m%5== 0)
      System.out.println (m);
    continue;
}
```

(i) Write the output of the program segment.

(ii) How many times does the body of the loop gets executed?

[ICSE 2016]

**Ans.** (i) 5  
10  
(ii) 3 times

**Q.10** Predict the output and also mention the number of times the loop runs.

```
int a=0;
while(a>-5)
{
    System.out.print(a+" ");
    System.out.print((--a*2));
    System.out.println();
    --a;
}
```

**Ans.** The output: 0    -2  
-2    -6  
-4    -10

Number of times the loop runs: 3



## EXERCISES

### I. Fill in the blanks:

1. When the statements are repeated sequentially a number of times in a program, the construct is known as .....
2. For loop is also known as ..... controlled loop.
3. .... loop is called an exit controlled loop.
4. .... loop executes at least once, if the condition is false.
5. .... loop checks the condition first before its execution.
6. To find the sum of any ten numbers, the loop will run ..... times.

### II. Answer the following questions:

1. What is 'for' loop? What are the parameters used in 'for' loop?
2. Define the following with their constructs:  
(a) Entry controlled loop (b) Exit controlled loop
3. Write down the syntax of:  
(a) do - while (b) while loop
4. What is the purpose of using  
(a) break statement (b) continue statement in a program?
5. Distinguish between while and do-while loop. [ICSE 2018]
6. What is meant by an infinite loop? Give an example. [ICSE 2008]
7. State one difference and one similarity between while loop and do-while loop. [ICSE 2010]

### III. Predict the output:

1. The following is a segment of a program.  

```
x = 1; y = 1;
if(n>0)
{
x = x + 1;
y = y +1;
}
```

What will be the value of x and y, if n assumes a value:  
(i) 1 (ii) 0 ? [ICSE 2009]
2. Analyze the following program segment and determine how many times the body of the loop will be executed (show the working).  

```
x = 5; y = 50;
while(x<=y)
{
y = y/x;
System.out.println(y);
}
```

 [ICSE 2009]
3. What will be the output of the following code?  

```
int m=2;
int n=15;
for(int i=1;i<5;i++)
m++;
-- n;
```



```
System.out.println("m="+m);
System.out.println("n="+n);
```

[ICSE 2010]

4. Analyze the following program segment and determine how many times the loop will be executed. What will be the output of the program segment?

```
int k=1,i=2;
while(++i<6)
k*=i;
System.out.println(k);
```

[ICSE 2010]

5. Give the output of the following program segment and also mention the number of times the loop is executed.

```
int a,b;
for(a=6;b=4;a<= 4; a=a+ 6)
{
if(a%b==0)
break;
}
System.out.println(a);
```

[ICSE 2017]

6. Give the output of the following program segment and also mention how many times the loop is executed.

```
int i;
for(i = 5; i >10; i ++ )
System.out.println( i );
System.out.println(i * 4);
```

[ICSE 2018]

#### IV. Rewrite the following programs:

1. Using for loop: [ICSE 2017]

```
int i=1;
int d=5;
do
{
d=d*2;
System.out.println(d);
i++;
}
while (i<=5);
```

2. Using while loop:

```
import java.util.*;
class Number
{
public static void main(String args[])
{
int n,r;
Scanner in = new Scanner(System.in);
System.out.println("Enter a number");
n=in.nextInt();
do
{
r=n%10;
n=n/10;
System.out.println(r);
}
while(n!=0);
}
}
```



## V. Unsolved Java Programs:

1. Write the programs in Java to display the first ten terms of the following series:
  - (a) 0, 1, 2, 3, 6, .....
  - (b) 1, -3, 5, -7, 9, .....
  - (c) 0, 3, 8, 15, .....
  - (d) 1, 11, 111, 1111, .....
  - (e) 1, 12, 123, 1234, .....
2. Write the programs in Java to find the sum of the following series:
  - (a)  $S = 1 + 1 + 2 + 3 + 5 + \dots$  to n terms
  - (b)  $S = 2 - 4 + 6 - 8 + \dots$  to n
  - (c)  $S = 1 + (1+2) + (1+2+3) + \dots + (1+2+3+ \dots + n)$
  - (d)  $S = 1 + (1*2) + (1*2*3) + \dots + (1*2*3* \dots * n)$
  - (e)  $S = 1 + \frac{1+2}{1*2} + \frac{1+2+3}{1*2*3} + \dots + \frac{1+2+3+ \dots + n}{1*2*3* \dots + n}$  [ICSE 2007]
3. Write the programs to find the sum of the following series:
  - (a)  $S = a + a^2 + a^3 + \dots + a^n$
  - (b)  $S = (a+1) + (a+2) + (a+3) + \dots + (a+n)$
  - (c)  $S = \frac{a}{2} + \frac{a}{5} + \frac{a}{8} + \frac{a}{11} + \dots + \frac{a}{20}$  [ICSE 2008]
  - (d)  $S = \frac{1}{a} + \frac{2}{a^2} + \frac{3}{a^3} + \dots$  to n
  - (e)  $S = a - a^3 + a^5 - a^7 + \dots$  to n
4. Write a program to enter two numbers and check whether they are co-prime or not.  
[Two numbers are said to be co-prime, if their HCF is 1 (one).]  
Sample Input: 14, 15  
Sample Output: They are co-prime.
5. Write a program to input a number. Display the product of the successors of even digits of the number entered by user.  
Input: 2745  
Output: 15  
[Hint: The even digits are: 2 and 4  
The product of successor of even digits is:  $3*5 = 15$ ]
6. 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.]  
Examples:  $12 = 3 * 4$   
 $20 = 4 * 5$   
 $42 = 6 * 7$  [ICSE 2018]
7. A prime number is said to be 'Twisted Prime', if the new number obtained after reversing the digits is also a prime number. Write a program to accept a number and check whether the number is 'Twisted Prime' or not.  
Sample Input: 167  
Sample Output: 761  
167 is a 'Twisted Prime'.



8. Write a program to input a number and check whether it is a prime number or not. If it is not a prime number then display the next number that is prime.  
Sample Input: 14  
Sample Output: 17
9. A number is said to be Duck if the digit zero is (0) present in it. Write a program to accept a number and check whether the number is Duck or not.  
The program displays the message accordingly. (The number must not begin with zero)  
Sample Input: 5063  
Sample Output: It is a Duck number.  
Sample Input: 7453  
Sample Output: It is not a Duck number.
10. Write a program that inputs number of runs made by a cricket player on each ball. Entering runs as -1 should display the message that player is out. Finally it displays the number of runs made and balls played by the player.
11. A computerised bus charges fare from each of its passengers based on the distance travelled as per the tariff given below:

Distance (in km)	Charges
First 5 km	₹ 80
Next 10 km	₹ 10/km
More than 15 km	₹ 8/km

As the passenger enters the bus, the computer prompts 'Enter distance you intend to travel'. On entering the distance, it prints his ticket and the control goes back for the next passenger. At the end of journey, the computer prints the following:

- (i) the number of passenger travelled
- (ii) total fare received

Write a program to perform the above task.

[Hint: Perform the task based on user controlled loop]

12. A special two-digit number is such that when the sum of its digits is added to the product of its digits, the result is equal to the original two-digit number.  
Example: Consider the number 59.  
Sum of digits =  $5 + 9 = 14$   
Product of digits =  $5 * 9 = 45$   
Sum of the sum of digits and product of digits =  $14 + 45 = 59$   
Write a program to accept a two-digit number. Add the sum of its digits to the product of its digits. If the value is equal to the number input, then display the message "Special two-digit number" otherwise, display the message "Not a special two-digit number". [ICSE 2014]
13. Write a program to input a number. Check and display whether it is a Niven number or not.  
(A number is said to be Niven which is divisible by the sum of its digits).  
Example: Sample Input 126  
Sum of its digits =  $1 + 2 + 6 = 9$  and 126 is divisible by 9. [ICSE 2016]
14. Write a program to accept a number and check whether it is a 'Spy Number' or not.  
(A number is spy if the sum of its digits equals the product of its digits.)  
Example: Sample Input: 1124  
Sum of the digits =  $1 + 1 + 2 + 4 = 8$   
Product of the digits =  $1*1*2*4 = 8$  [ICSE 2017]





15. You can multiply two numbers 'm' and 'n' by repeated addition method.  
 For example,  $5 * 3 = 15$  can be performed by adding 5 three times  $\Rightarrow 5 + 5 + 5 = 15$   
 Similarly, successive subtraction of two numbers produces 'Quotient' and 'Remainder' when a number 'a' is divided by 'b' ( $a > b$ ).  
 For example,  $5/2 \Rightarrow$  Quotient = 2 and Remainder = 1  
 Follow steps shown below:

Process	Result	Counter
5 - 2	3	1
3 - 2	1	2

- Sample Output: The last counter value represents 'Quotient'  $\Rightarrow 2$   
 The last result value represents 'Remainder'  $\Rightarrow 1$   
 Write a program to accept two numbers. Perform multiplication and division of the numbers as per the process shown above by using switch case statement.
16. Using a switch statement, write a menu driven program to:
- (a) generate and display the first 10 terms of the Fibonacci series  
 0, 1, 1, 2, 3, 5 .....  
 The first two Fibonacci numbers are 0 and 1, and each subsequent number is the sum of the previous two.
  - (b) find the sum of the digits of an integer that is input by the user.  
 Sample Input: 15390  
 Sample Output: Sum of the digits = 18
- For an incorrect choice, an appropriate error message should be displayed.



## Unit IX: Nested Loop

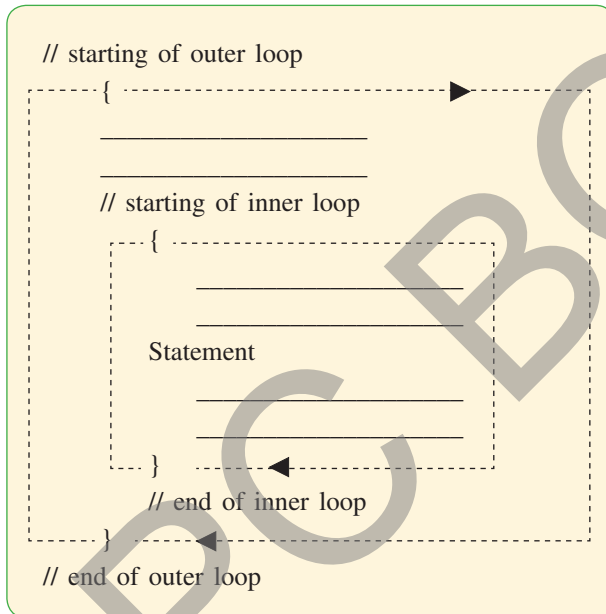
### LEARNING SCOPE

- Definition and construct of a Nested Loop
- Types of Nested Loops
- Break statement in a nested loop
- Labelled break statement
- Continue statement and Labelled continue in nested loop
- Solved programs based on nested loop
- Triangular Pattern (Right angled triangle only) and Rectangular Pattern.

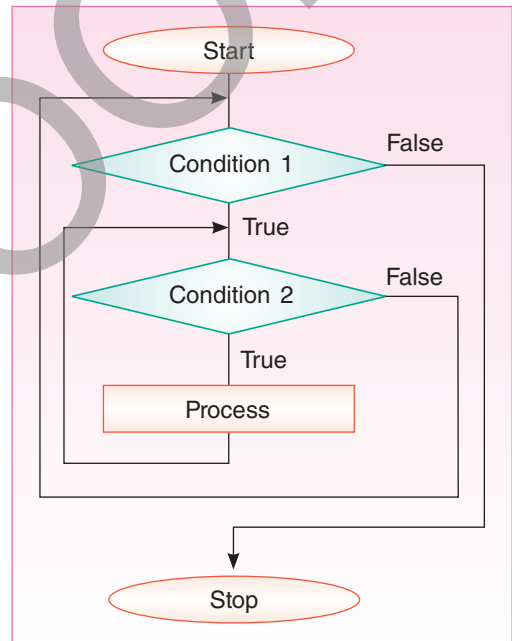
In the previous unit, you have learnt about looping constructs in which a statement/a set of statements repeats for a given number of times. When you use a loop within another loop is said to be a nested loop.

In the nested loop, you will notice that the inner loop repeats a number of times for each repetition of the outer loop.

The construct and flow chart of a nested loop is illustrated as:



*Construct of a Nested Loop*



*Flow chart of a Nested Loop*

### Types of Nested Loops

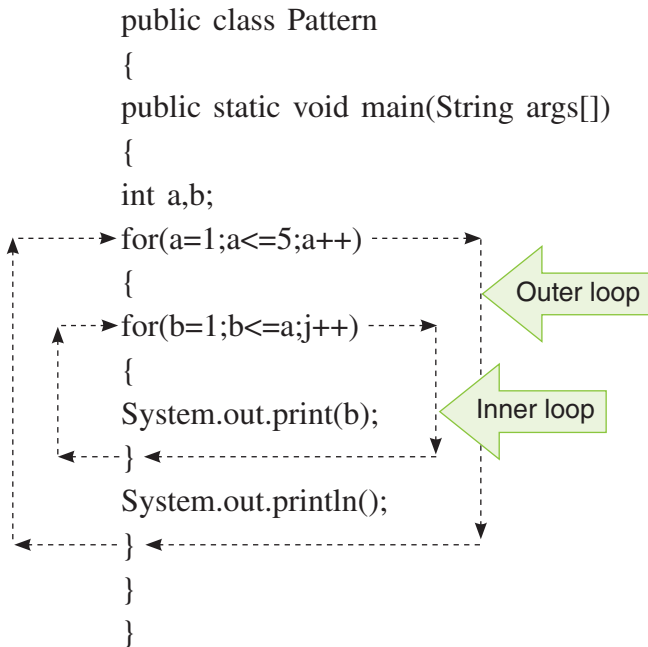
A nested loop can be designed in the following ways:

1. Nested for loop
2. Nested while loop
3. Nested do-while loop

#### 1. Nested for loop

When a 'for' loop is used within another 'for' loop, is called a nested for loop.

A sample program is illustrated as:



#### Output

```
1
12
123
1234
12345
```

With reference to the above program:

Can you guess how many times the loop runs to produce the output?

Here, the execution takes place as stated below:

Number of times the outer loop runs=5

For an inner loop:

When a=1: the number of times inner loop runs = 1 (as updated value of b=1)

When a=2: the number of times inner loop runs = 2 (as updated value of b=2)

When a=3: the number of times inner loop runs = 3 (as updated value of b=3)

When a=4: the number of times inner loop runs = 4 (as updated value of b=4)

When a=5: the number of times inner loop runs = 5 (as updated value of b=5)

Thus, total number of iterations= 5+4+3+2+1= 15

#### Note:

It is important to know that the inner loop is closed first and then the outer loop is closed. Similarly, the statements of inner loop are executed first which is followed by the outer loop.

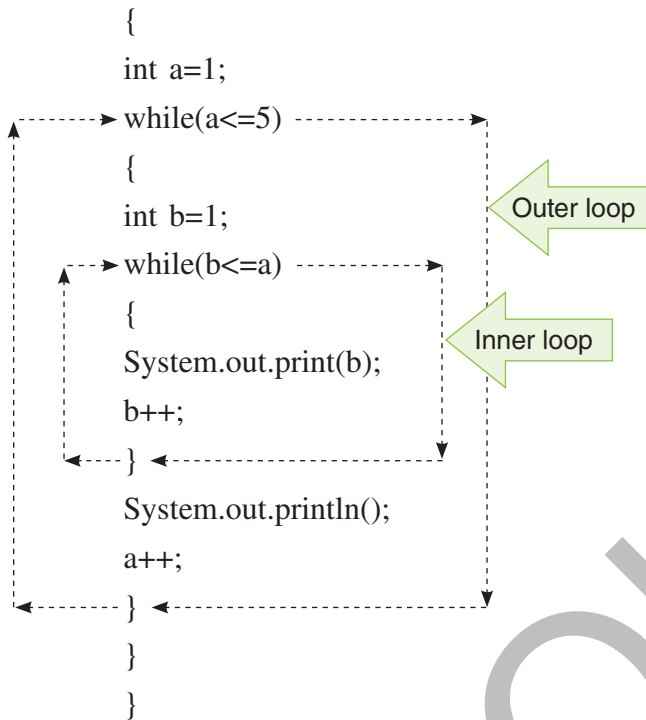
## 2. Nested while loop

When a 'while' loop is used within another 'while' loop, is called a nested while loop. An example is illustrated.

//A sample program is illustrated as:

```
public class Pattern
{
    public static void main(String args[])
    {
```





#### Output

```

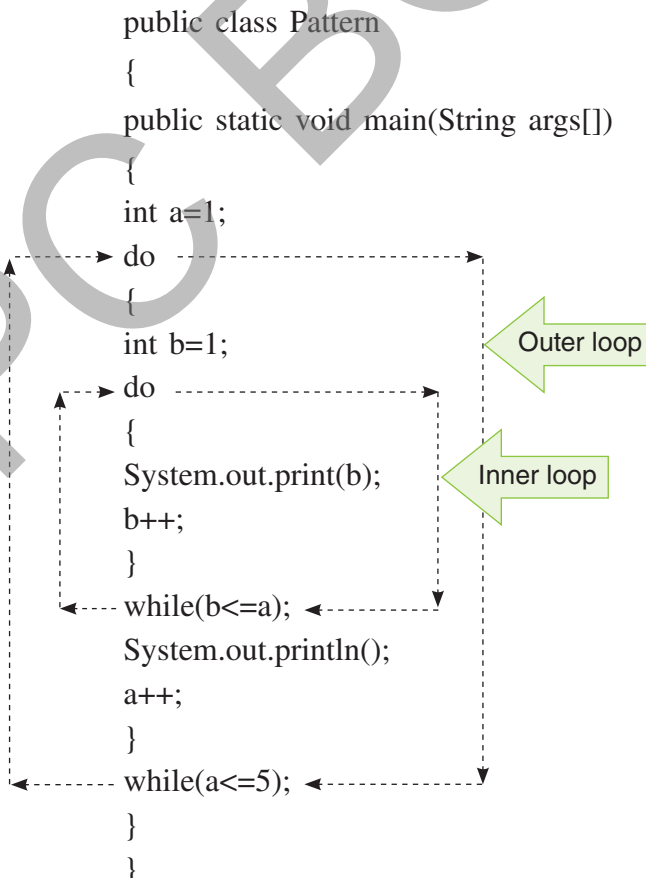
1
12
123
1234
12345

```

### 3. Nested do-while loop

When a 'do-while' loop is used within another 'do-while' loop, is called a nested do-while loop.

//A sample program is illustrated as:



#### Output

```

1
12
123
1234
12345

```

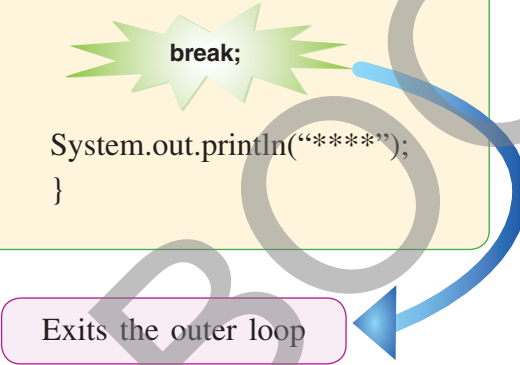
## Break statement in a nested loop

This statement is used for unusual termination of a block. As soon as, the '*break*' statement is invoked, the control will come out of the block by ignoring rest of the statements. The break statement can be used in the inner loop as well as in the outer loop.

### Use of break statement in the outer loop

The example illustrates the use of 'break' statement in the outer loop:

```
int i,j;
for(i=1;i<=5;i++)
{
    for(j=1;j<=3;j++)
    {
        System.out.println(i*j);
    }
    if(i> 2)
        break;
    System.out.println("*****");
}
```

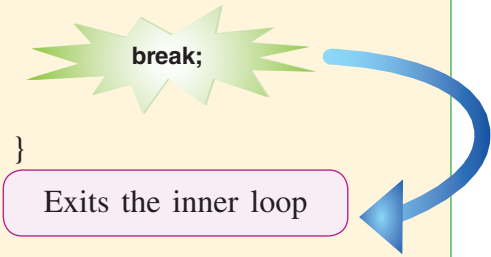


Exits the outer loop

### Use of break statement in the inner loop

This example illustrates the use of break in the inner loop.

```
int i,j;
for(i=1;i<=5;i++)
{
    for(j=1;j<=3;j++)
    {
        System.out.println(i*j);
        if(i*j> 8)
            break;
    }
}
```

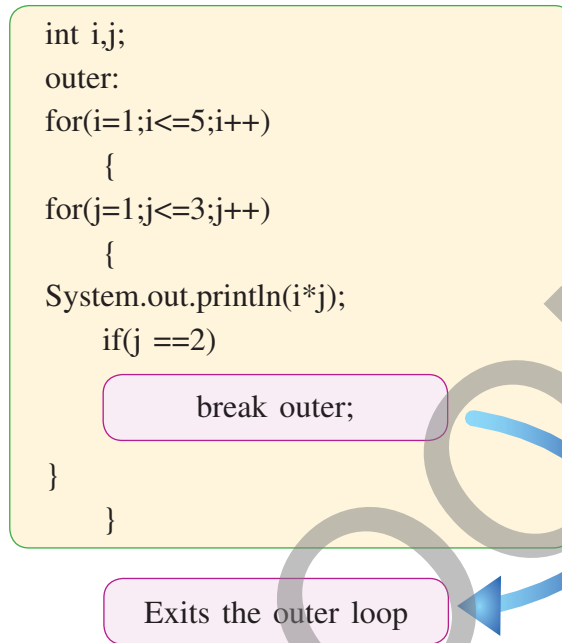


Exits the inner loop



## Labelled Break statement

Sometimes, it may happen that the requirement of the user is fulfilled during the iteration of the inner loop and so statements in the outer loop should not be executed further. Labelled break statement will be helpful to serve this purpose. The termination of outer loop from the block of inner loop can be done by using keyword 'break' along with a label (say, 'outer') as in the illustration shown below:



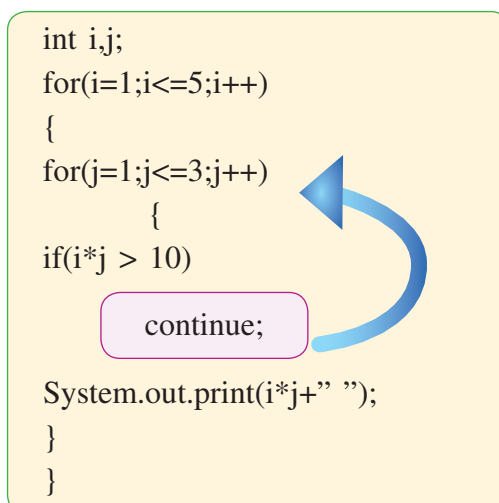
## Continue Statement in Nested Loop

Like break statement, you can also use 'continue' statement in the nested loop to get your task done. As soon as, the control reaches to 'continue', it takes the control to the next iteration by ignoring rest of the statements. An example is illustrated as follows:

### Continue statement used in Inner Loop

Here, you can notice when the control reaches 'continue', it immediately takes the control to the next iteration of the inner loop by ignoring remaining statements of the inner block.

In the example shown below, the '*continue*' statement is used in the inner loop.

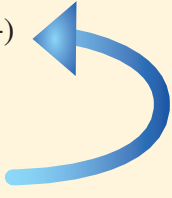


### Continue statement used in Outer Loop

When 'continue' is used in the outer loop then the control returns to the next iteration by ignoring rest part of the statements.

An example is illustrated below to show the use of 'continue' statement in the outer loop.


```
int i,j;
for(i=1;i<=5;i++)
{
    if(i == 2)
        continue;
    for(j=1;j<=3;j++)
    {
        System.out.print(i*j+" ");
    }
}
```



### Labelled Continue Statement

'Labelled continue' is a situation where you can skip the iterations of the inner loop. This task can be done by using the keyword 'continue outer' in the inner loop by ignoring the rest of the statements.

```
int i,j;
outer:
for(i=1;i<=5;i++)
{
    for(j=1;j<=3;j++)
    {
        if(i%2==0)
            continue outer;
        System.out.print(i*j+"");
    }
}
```



## SOLVED PROGRAMS ON NESTED LOOPS

**Prog. 1** Write a program to find the sum of the series:

$$S = 1 + \frac{1}{2!} - \frac{1}{3!} + \frac{1}{4!} + \dots \text{ to } \frac{1}{n!}$$

```
// To find the sum of the given series
import java.util.*;
public class Series
{
public static void main (String args[])
{
Scanner in=new Scanner(System.in);
int a,b,n;
double s=0;
System.out.println("Enter the value of n");
n=in.nextInt();
for(a=1;a<=n;a++)
{
int f=1;
for(b=1;b<=a;b++)
{
f=f*b;
}
s=s+(double)1/f;
}
System.out.println("The sum of the series = "+s);
}
}
```

**Prog. 2** Write a program to display all the prime numbers between 1 and 100. (Prime number is a number which is divisible by 1 and the number itself only.)

```
// To display all prime numbers between 1 and 100
public class Prime
{
public static void main (String args[])
{
int a,b;
System.out.println("The prime numbers between 1 and 100 are:");
for(a=1;a<=100;a++)
```





```

{
int c=0;
for(b=1;b<=a;b++)
{
if(a%b==0)
c=c+1;
}
if(c==2)
System.out.println(a);
}
}
}

```

### Prog. 3

Write a program to input a number and print whether the number is a 'special number' or not. A number is said to be special number if the sum of the factorial of the digits of the number is the same as the original number.

Sample Input: 145 is a special number because  $1! + 4! + 5! = 1 + 24 + 120 = 145$  where, ! stands for the factorial of the number and the factorial value of a number is the product of all integers from 1 to that number. e.g.,  $5! = 1*2*3*4*5 = 120$  [ICSE 2011]

// To check special number using Scanner Class

```

import java.util.*;
public class Special
{
public static void main(String args[])
{
int m,n,i,d,f=1,s=0;
Scanner in=new Scanner(System.in);
System.out.println("Enter a number :");
n=in.nextInt();
m=n;
while(m!=0)
{
d=m%10;
for(i=1;i<=d;i++)
f=f*i;
s=s+f;
f=1;
m=m/10;
}
if(s==n)

```



```

{
System.out.println("The sum of factorial of digits =" +s);
System.out.println(n + " is a special number ");
}
else
{
System.out.println("The sum of factorial of digits =" +s);
System.out.println(n+ " is not a special number ");
}
}
}
}

```

**Prog. 4** Write a program to display twenty simultaneous 'Prime Numbers', starting from a given number, entered by the user.  
Sample Input : 8  
Sample Output : 11, 13, 17, 19, .....

```

// To display 20 prime numbers
import java.util.*;
public class Twenty_Prime
{
public static void main (String args[])
{
Scanner in=new Scanner(System.in);
int a,b,c,n,k=1;
System.out.println("Enter the first number");
n=in.nextInt();
System.out.println("Twenty prime numbers from "+ n +":");
do
{
c=0;
for(a=1;a<=n;a++)
{
if(n%a==0)
c=c+1;
}
if(c==2)
{
System.out.println(n+ " ");
k=k+1;
}
n=n+1;
}
}

```



```
while(k<=20);
}
}
```

**Prog. 5**

A happy number is defined as:

Take a positive number and replace the number by the sum of the squares of its digits. Repeat the process until the number equals 1 (one). If the number ends with 1 then it is called a happy number. Write a program to input a number and check whether it a 'Happy Number' or not. The program displays a message accordingly.

Sample Input : 31

Solution :  $31 \Rightarrow 3^2 + 1^2 = 10 \Rightarrow 1^2 + 0 = 1$

Sample Output : A Happy Number

```
// To check Happy number
import java.util.*;
class Happy_Number
{
public static void main(String args[])
{
Scanner in = new Scanner(System.in);
int num,sum=0,d;
System.out.println("Enter a number");
num = in.nextInt();
sum=num;
do
{
num=sum;sum=0;
do
{
d=num%10;
sum=sum+d*d;
num=num/10;
}
while(num>0);
}
while(sum>9);
if(sum==1)
System.out.println("Happy Number");
else
System.out.println("Not a Happy Number");
}
}
```



**Prog. 6**

The International Standard Book Number (ISBN) is a unique numeric book identifier which is printed on every book. The ISBN is a 10 digit code. The ISBN is legal if:

$1 \times \text{digit}_1 + 2 \times \text{digit}_2 + 3 \times \text{digit}_3 + 4 \times \text{digit}_4 + 5 \times \text{digit}_5 + 6 \times \text{digit}_6 + 7 \times \text{digit}_7 + 8 \times \text{digit}_8 + 9 \times \text{digit}_9 + 10 \times \text{digit}_{10}$  is divisible by 11.

Example: For an ISBN 1401601499

Sum =  $1 \times 1 + 2 \times 4 + 3 \times 0 + 4 \times 1 + 5 \times 6 + 6 \times 0 + 7 \times 1 + 8 \times 4 + 9 \times 9 + 10 \times 9$   
= 253 which is divisible by 11.

(i) Input the ISBN code as a 10 digit integer.

(ii) If the ISBN is not a 10 digit integer, output the message, “Illegal ISBN” and terminate the program.

(iii) If the number is 10 digits, extract the digits of the number and compute the sum as explained above.

If the sum is divisible by 11, output the message, “Legal ISBN”. If the sum is not divisible by 11, output the message, “Illegal ISBN”.

[ICSE 2013]

// A program on the International Standard Book Number (ISBN)

```
import java.util.*;
```

```
class ISBN
```

```
{
```

```
public static void main(String args[])
```

```
{
```

```
Scanner in = new Scanner(System.in);
```

```
int n,n1,i,r,sum=0,c=0;
```

```
System.out.println(“Enter a 10 digit code”);
```

```
n=in.nextInt();
```

```
n1=n;
```

```
while(n1>0)
```

```
{
```

```
n1=n1/10;
```

```
c++;
```

```
}
```

```
if(c<10 || c>10)
```

```
System.out.println(“Illegal ISBN”);
```

```
else
```

```
{
```

```
for(i=10;i>=1;i--)
```

```
{
```

```
r=n%10;
```

```
sum=sum+r*i;
```

```
n=n/10;
```

```
}
```



```

if(sum%11==0)
System.out.println("A valid ISBN code");
else
System.out.println("Not a valid ISBN code");
}
}
}

```

<b>Prog. 7</b>	Write a program to display the Floyd's Triangle: [ICSE 2016]
----------------	--

```

1
2 3
4 5 6
7 8 9 10
11 12 13 14 15

```

```

// To display Floyd's Triangle
public class Pattern
{
public static void main(String args[])
{
int a,b,p=0;
for(a=1;a<=5;a++)
{
for(b=1;b<=a;b++)
{
p=p+1;
System.out.print(p+" ");
}
System.out.println();
}
}
}

```

<b>Prog. 8</b>	Write a program in Java to display the given pattern
----------------	--

```

1 2 3 4 5 6 7
1 2 3 4 5
1 2 3
1

```

```

// To display the given pattern
public class Pattern
{
public static void main(String args[ ])
{

```



```
int i,j;
for(i=7;i>=1;i=i-2)
{
for(j=1;j<=i;j++)
System.out.print(j+" ");
System.out.println();
}
}
}
```

<b>Prog. 9</b>	Write a program to display the given pattern: <div>5 4 3 2 1 4 3 2 1 3 2 1 2 1 1</div>
----------------	---

```
//To display the given pattern
public class Pattern
{
public static void main(String args[])
{
int i,j,k,p=1;
System.out.println("The Pattern:");
for(i=5;i>=1;i--)
{
for(k=1;k<=p;k++)
System.out.print(" "); //To print a blank
for(j=i;j>=1;j--)
System.out.print(j);
System.out.println();
p=p+1;
}
}
}
```

<b>Prog. 10</b>	Write a program to display the given pattern: <div>1 2 3 4 5 2 2 3 4 5 3 3 3 4 5 4 4 4 4 5 5 5 5 5 5</div>
-----------------	---

```
// To display the given pattern
```



```

public class Pattern
{
public static void main(String args[])
{
int a,b,c,p=2;
for(a=1;a<=5;a++)
{
for(b=1;b<=a;b++)
System.out.print(a);
for(c=p;c<=5;c++)
System.out.print(c);
System.out.println();
p=p+1;
}
}
}

```

**Prog. 11** Write a program in Java to display the given pattern.

```

1 2 3 4 5 5 4 3 2 1
1 2 3 4      4 3 2 1
1 2 3          3 2 1
1 2              2 1
1                  1

```

```

// To display the given pattern
public class Special_Pattern
{
public static void main(String args[ ])
{
int i,j,k,p,d;k=1;p=5;d=1;
System.out.println("The pattern of Triangle:");
for(i=1;i<=5;i++)
{
for(j=1;j<=p;j++)
System.out.print(j);
for(k=1;k<=d;k++)
System.out.print(" ");
for(k=p;k>=1;k--)
System.out.print(k);
System.out.println();
p=p-1;d=d+2;
}
}
}

```



}

**Prog. 12** Write a program in Java to display the given pattern.

```
1 A A A A A
2 2 B B B B
3 3 3 C C C
4 4 4 4 D D
5 5 5 5 5 E
```

```
// To display the given pattern
public class Pattern
{
    public static void main(String args[])
    {
        int a,b,c,d,p,k;p=65;k=5;
        System.out.println("The pattern:");
        for(a=1;a<=5;a++)
        {
            for(b=1;b<=a;b++)
            System.out.print(a+" ");
            for(c=k;c>=1;c--)
            {
                System.out.print((char)p+" ");
            }
            k=k-1;
            p=p+1;
            System.out.println();
        }
    }
}
```





REVIEW **Insight**

**Q.1** What is meant by 'labelled break' statement?  
**Ans.** It is a method of terminating the outer loop when the condition of inner loop satisfies. It is done by using a keyword 'break outer' in the inner loop.

**Q.2** Find the error:  

```
int j,k;
for(k=1;k<=10;k--)
{
    for(j=1;j>=k;j++)
        System.out.println(j);
}
```

**Ans.** In the outer loop, the update expression should be k++ and in the inner loop, the condition should be j<=k.

**Q.3** Give two differences between 'break' and 'continue'.

**Ans.** The two differences between break and continue are:

Break	Continue
1. Break statement is used to terminate a block in which it exists.	1. Continue statement is used to move the control to the next iteration of the loop ignoring other statements.
2. Break statement can be used in switch-case as well as in loop statements.	2. Continue statement is used only in the loop statement.

**Q.4** Predict the output:  

```
int i,j;
for(j=1;j<=5;j+=2)
{
    for(k=1;k<=j;k++)
        System.out.print(k);
    System.out.println();
}
```

**Ans.**

1  
123  
12345

**Q.5** Study the snippet given below and answer the following questions:

```
int j,k,p=-1;
for(j=-2;j<=1;j++)
{
    for(k=j;k<=0;k++)
        k=Math.max(j*k,p);
    System.out.print(k);
    p=p+2;
}
```

**Ans.** (i) How many times will the outer loop run? (ii) Predict the output.  
(i) The outer loop runs 4 times. (ii) The output: 5 2 4 1

**Q.6** Predict the output:  

```
int k=5,j;
while(k>=1)
{
    j=1;
    while(j<=k)
    {
```



**Ans.**

## NOTES

## EXERCISES

### I. Write whether the following statements are True/False:

1. Nested loop contains a single loop.
2. When break statement is applied, it terminates the loop.
3. The outer loop follows next iteration when iterations of inner loop is over.
4. Nested loop means the using of two or more loops in a program.
5. Labelled break statement allows the next iteration of the loop from any place of looping structure.
6. In a nested loop, break and continue can be used simultaneously.

### II. Fill in the blanks:

1. .... break is used to terminate an outer loop.
2. A loop within another loop is called .....
3. Termination of ..... loop takes place before ..... loop.
4. .... statement will repeat a loop for the next iteration after ignoring some statements of the loop.
5. .... statement terminates a switch-case statement as well as a loop.

### IV. Answer the following questions:

1. What is a nested loop?
2. Write down the syntax of a nested for loop.
3. What action will you take to terminate an outer loop from the block of an inner loop?
4. Give two differences between 'Break' and 'Continue'.
5. What is significance of 'break outer' and 'continue outer' in a nested loop?
6. Write down the constructs (syntax) of:
  - (a) Nested do-while loop
  - (b) Nested while loop

### V. Unsolved Programs:

1. Write programs to find the sum of the following series:

(a)  $S = 1 + \frac{3}{2!} + \frac{5}{3!} + \frac{7}{4!} + \dots$  to n

(b)  $S = a + \frac{a}{2!} + \frac{a}{3!} + \frac{a}{4!} + \dots + \frac{a}{n!}$

(c)  $S = a - \frac{a}{2!} + \frac{a}{3!} - \frac{a}{4!} + \dots$  to n

(d)  $S = \frac{a}{2!} - \frac{a}{3!} + \frac{a}{4!} - \frac{a}{5!} + \dots + \frac{a}{10!}$

(e)  $S = \frac{2}{a} + \frac{3}{a^2} + \frac{5}{a^3} + \frac{7}{a^4} + \dots$  to n



2. Write a program to input two numbers and check whether they are twin prime numbers or not.  
**Hint:** Twin prime numbers are the prime numbers whose difference is 2.  
 For example: (5,7), (11,13), ..... and so on.
3. Write a program to display all the numbers between 100 and 200 which don't contain zeros at any position.  
 For example: 111, 112, 113, ....., 199
4. Write a program to display all prime palindrome numbers between 10 and 1000.  
**[Hint:** A number which is prime as well a palindrome is said to be 'Prime Palindrome' number.]  
 For example: 11, 101, 131, 151, .....
5. In an entrance examination, students have been appeared in English, Maths and Science papers. Write a program to calculate and display average marks obtained by all the students. Take number of students appeared and marks obtained in all three subjects by every student along with the name as inputs.  
 Display the name, marks obtained in three subjects and the average of all the students.
6. Write a program in Java to enter a number containing three digits or more.  
 Arrange the digits of the entered number in ascending order and display the result.  
 Sample Input: Enter a number 4972  
 Sample Output: 2, 4, 7, 9
7. Write a program to input a number and check whether it is 'Magic Number' or not.  
 Display the message accordingly.  
 A number is said to be a magic number if the eventual sum of digits of the number is one.  
 Sample Input : 55  
 Then,  $5 + 5 = 10$ ,  $1 + 0 = 1$   
 Sample Output: Hence, 55 is a Magic Number.  
 Similarly, 289 is a Magic Number.
8. A number is said to be Multiple Harshad number, when divided by the sum of its digits, produces another 'Harshad Number'. Write a program to input a number and check whether it is a Multiple Harshad Number or not.  
 (When a number is divisible by the sum of its digit, it is called 'Harshad Number').  
 Sample Input: 6804  
**Hint.**  $6804 \Rightarrow 6+8+0+4 = 18 \Rightarrow 6804/18 = 378$   
 $378 \Rightarrow 3+7+8 = 18 \Rightarrow 378/18 = 21$   
 $21 \Rightarrow 2+1 = 3 \Rightarrow 21/3 = 7$   
 Sample Output: Multiple Harshad Number
9. Write the programs to display the following patterns:
 

<p>(a)</p> <pre> 1 3 1 5 3 1 7 5 3 1 9 7 5 3 1           </pre> <p>(d)</p> <pre> 1 1 0 1 0 1 1 0 1 0 1 0 1 0 1           </pre>	<p>(b)</p> <pre> 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15           </pre> <p>(e)</p> <pre> 5 5 5 5 5 4 4 4 4 3 3 3 2 2 1           </pre>	<p>(c)</p> <pre> 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1           </pre> <p>(f)</p> <pre> 1 2 3 4 5 2 2 3 4 5 3 3 3 4 5 4 4 4 4 5 5 5 5 5 5           </pre>
---	---	---



(g)
 

```

*
* #
* # *
* # * #
* # * # *
```

(h)
 

```

5 4 3 2 1
5 4 3 2
5 4 3
5 4
5
```

(i)
 

```

1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

10. Write a program to generate a triangle or an inverted triangle till n terms based upon the user’s choice. [ICSE 2009]

<b>Example 1:</b>	<b>Example 2:</b>
Input: Type 1 for a triangle and Type 2 for an inverted triangle Enter your choice 1 Enter the number of terms 5	Input: Type 1 for a triangle and Type 2 for an inverted triangle Enter your choice 2 Enter the number of terms 6
<b>Sample Output:</b> <pre> 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5</pre>	<b>Sample Output:</b> <pre> 6 6 6 6 6 6 5 5 5 5 5 4 4 4 4 3 3 3 2 2 1</pre>

11. Using the switch statement, write a menu driven program for the following: [ICSE 2016]

- (a) To print the Floyd’s triangle:
 

```

1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```
- (b) To display the following pattern:
 

```

I
I C
I C S
I C S E
```

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

