Subject - Java

Class – BCA Final

Pro Surinder Kaur

Introduction

Java is a popular programming language, created in 1995.

It is owned by Oracle, and more than 3 billion devices run Java.

It is used for:

- Mobile applications (specially Android apps)
- Desktop applications
- Web applications
- Web servers and application servers
- Games
- Database connection
- And much, much more!

Java Variables

Variables are containers for storing data values.

In Java, there are different types of variables, for example:

1. String - stores text, such as "Hello". String values are surrounded by double quotes

2. Int - stores integers (whole numbers), without decimals, such as 123 or -123

3. Float - stores floating point numbers, with decimals, such as 19.99 or -19.99

4. Char - stores single characters, such as 'a' or 'B'. Char values are surrounded by single quotes

5. Boolean - stores values with two states: true or false

Java Data Types

- Data types are divided into two groups:

- Primitive data types - includes byte, short, int, long, float, double, boolean and char

- Non-primitive data types - such as String, Arrays and Classes (you will learn more about these in a later chapter)

Primitive Data Types

Data Type	Size	Description
Byte	1 byte	Stores whole numbers from -128 to 127
Short	2 bytes	Stores whole numbers from -32,768 to 32,767
Int	4 bytes	Stores whole numbers from -2,147,483,648 to 2,147,483,647
Long	8 bytes	Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
Float	4 bytes	Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits
Double	8 bytes	Stores fractional numbers. Sufficient for storing 15 decimal digits
Boolean	1 bit	Stores true or false values
Char	2 bytes	Stores a single character/letter

Non-Primitive Data Types

Non-primitive data types are called **reference types** because they refer to objects.

The main difference between **primitive** and **non-primitive** data types are:

- Primitive types are predefined (already defined) in Java. Non-primitive types are created by the programmer and is not defined by Java (except for String).
- Non-primitive types can be used to call methods to perform certain operations, while primitive types cannot.
- A primitive type has always a value, while non-primitive types can be null.
- A primitive type starts with a lowercase letter, while non-primitive types starts with an uppercase letter.
- The size of a primitive type depends on the data type, while non-primitive types have all the same size.

Java Operators

Operators are used to perform operations on variables and values.

Java divides the operators into the following groups:

1. Arithmetic operators

2. Assignment operators

3. Comparison operators

4. Logical operators

5. Bitwise operators

Arithmetic Operators

Arithmetic operators are used to perform common mathematical operations.

Operator	Name	Description
+	Addition	Adds together two values
-	Subtraction	Subtracts one value from another
*	Multiplication	Multiplies two values
/	Division	Divides one value by another
%	Modulus	Returns the division remainder
++	Increment	Increases the value of a variable by 1
--	Decrement	Decreases the value of a variable by 1

Java Assignment Operators

Assignment operators are used to assign values to variables.

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
&=	x &= 3	x = x & 3
|=	x |= 3	x = x | 3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3

Java Comparison Operators

Comparison operators are used to compare two values (or variables).

Operator	Name	Example
==	Equal to	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Java Logical Operators

You can also test for true or false values with logical operators.

<table>
<tr><th>Operator</th><th>Name</th><th>Description</th><th>Example</th></tr>
<tr><td>&&</td><td>Logical and</td><td>Returns true if both statements are true</td><td>x < 5 && x < 10</td></tr>
<tr><td>||</td><td>Logical or</td><td>Returns true if one of the statements is true</td><td>x < 5 || x < 4</td></tr>
<tr><td>!</td><td>Logical not</td><td>Reverse the result, returns false if the result is true</td><td>!(x < 5 && x < 10)</td></tr>
</table>

Bitwise Operators

Bitwise operators are used to performing the manipulation of individual bits of a number.

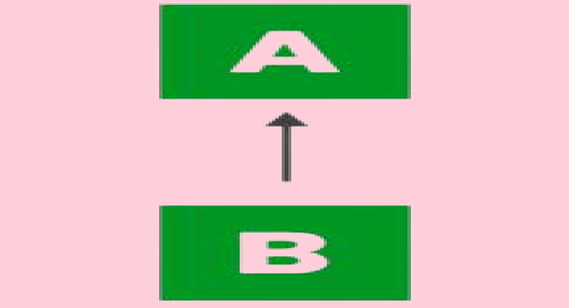
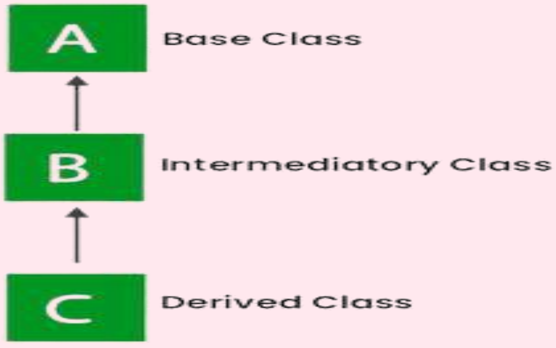
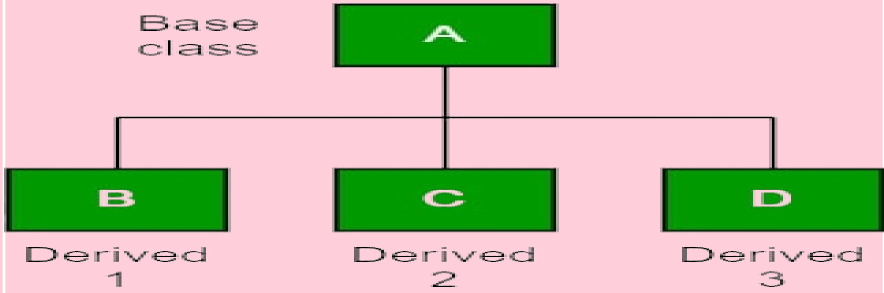
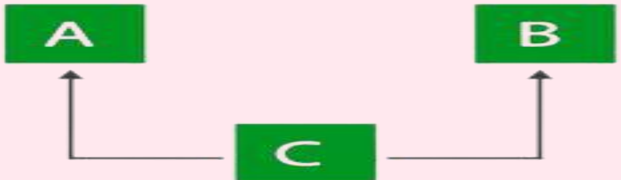
Name	Description
Bitwise OR (|)	This operator is a binary operator, denoted by ‘|’. It returns bit by bit OR of input values.
Bitwise AND (&)	This operator is a binary operator, denoted by ‘&.’ It returns bit by bit AND of input values.
Bitwise XOR (^)	This operator is a binary operator, denoted by ‘^.’ It returns bit by bit XOR of input values.

Inheritance

In Java, it is possible to inherit attributes and methods from one class to another.

Java Inheritance Types

1. Single Inheritance
2. Multilevel Inheritance
3. Hierarchical Inheritance
4. Multiple Inheritance
5. Hybrid Inheritance

Name	Explain	Figure
Single Inheritance	In single inheritance, subclasses inherit the features of one superclass.	 Single Inheritance
Multilevel Inheritance	In Multilevel Inheritance, a derived class will be inheriting a base class, and as well as the derived class also acts as the base class for other classes.	 A Base Class B Intermediatory Class C Derived Class Multilevel Inheritance
Hierarchical Inheritance	In Hierarchical Inheritance, one class serves as a superclass (base class) for more than one subclass.	 Base class A B C D Derived 1 Derived 2 Derived 3
Multiple Inheritance	In Multiple Inheritance one class can have more than one superclass and inherit	

Java - Classes and Objects.

Classes and objects are the two main aspects of object-oriented programming.

Look at the following illustration to see the difference between class and objects:

Example

Class	Objects
Fruit	Apple Mango Banana

Another Example

Class	Objects
Car	Volvo Audi Toyota

So, a class is a template for objects, and an object is an instance of a class. When the individual objects are created, they inherit all the variables and methods from the class.

Java Constructors

Java constructors or constructors in Java is a terminology used to construct something in our programs. A constructor in Java is a special method that is used to initialize objects. The constructor is called when an object of a class is created. It can be used to set initial values for object attributes.

Types of Constructors

- Default Constructor
- Parameterized Constructor
- Copy Constructor

Name	Explain
Default Constructor	A constructor that has no parameters is known as default the constructor. A default constructor is invisible. And if we write a constructor with no arguments, the compiler does not create a default constructor. It is taken out. It is being overloaded and called a parameterized constructor. The default constructor changed into the parameterized constructor. But Parameterized constructor can’t change the default constructor.
Parameterized Constructor	A constructor that has parameters is known as parameterized constructor. If we want to initialize fields of the class with our own values, then use a parameterized constructor.
Copy Constructor	Unlike other constructors copy constructor is passed with another object which copies the data available from the passed object to the newly created object.