Java Introduction

a. Java is a high level, robust, secured and object-oriented programming language. Structured in terms of classes, in which the members of the classes are treated as one single group.
b. Java support for multiple references for same object

c. **NOTE**: One reference can point to only one object for a time. Hence it is impossible to point to multiple objects with the same reference

2. Java Programming Styles
    a. Java Application - The programs that you execute in the command prompt
    b. Java Applets - Applets are run on the web browser or applet viewer

3. Java Program
    a. Any plain text editor or text editor capable of saving in ASCII format can be use to create a source file.e.g. DOS EDIT, Notepad. should be saved with a .java extension.

4. List of Java keywords and Escape characters and Packages

	Keyword		Keyword
1.	Abstract	26.	interface
2.	Assert	27.	long
3.	boolean	28.	native
4.	break	29.	new
5.	byte	30.	package
6.	case	31.	private
7.	catch	32.	protected
8.	char	33.	public
9.	class	34.	return
10.	continue	35.	short
11.	default	36.	static
12.	do	37.	strictfp
13.	double	38.	super
14.	else	39.	switch
15.	enum	40.	synchronized
16.	extends	41.	this
17.	final	42.	throw
18.	finally	43.	throws
19.	float	44.	transient
20.	for	45.	try
21.	if	46.	void
22.	implements	47.	volatile
23.	import	48.	while
24.	instanceof	49.	const
25.	int	50.	goto

1.	True	Not Keyword
2.	False	Not Keyword
3.	Null	Not Keyword

4. Escape characters in java

Escape Sequence	Description
\t	Insert a tab in the text at this point.
\b	Insert a backspace in the text at this point.
\n	Insert a newline in the text at this point.
\r	Insert a carriage return in the text at this point.
\f	Insert a formfeed in the text at this point.
\'	Insert a single quote character in the text at this point.
\"	Insert a double quote character in the text at this point.
\\	Insert a backslash character in the text at this point.

5. Core packages in Java

Package	Description
java.lang	basic language functionality and fundamental types
java.util	collection data structure classes
java.io	file operations
java.math	multiprecision arithmetics
java.nio	the Non-blocking I/O framework for Java
java.net	networking operations, sockets, DNS lookups, ...
java.security	key generation, encryption and decryption
java.sql	Java Database Connectivity (JDBC) to access databases
java.awt	basic hierarchy of packages for native GUI components
java.text	Provides classes and interfaces for handling text, dates, numbers, and messages in a manner independent of natural languages.
java.rmi	Provides the RMI package.
java.time	The main API for dates, times, instants, and durations.
java.beans	The java.beans package contains classes and interfaces related to JavaBeans components.

OO Concepts

Classes and Objects

class is a template for multiple objects with similar features.

collection of classes is called a program

Object is an instance of a class

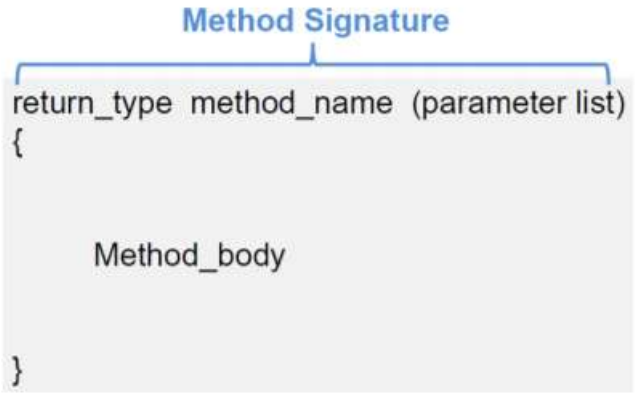
dot operator is used to access the values of the variables and to call the methods of the class

Instance and Static Variables

Instance - characteristic take different values for rach student. Hence they are declared in the class, but outside the methods

Static - If the same value of an attribute is shared by all the class instances are named as class variables and are defined as “static”. They are declared in the classes, but outside the methods

Methods



(@, #, %, etc) are not included in the method names

There are two major types of methods.

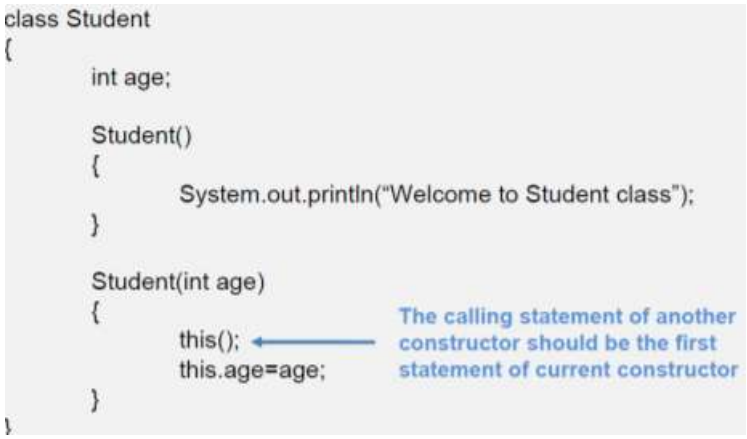
- Instance Methods
- Class Methods
  - Methods declared with static (class methods) cannot use ‘this’ keyword.
  - not allowed to access the instance variables
  - Access Modifiers can be used
  - main method is also a class method

Method Overloading

- allows to have more than one method with the same name, but different method signatures

Constructor Overloading

- constructor can call another constructor using “this” keyword.



Operators, Access Levels, Control Statements

Operators

Arithmetic Operators

Operator	Meaning	Example
+	Addition	8+10
-	Subtraction	10-5
*	Multiplication	5*3
/	Division	90/4
%	Modulus	10%6

Assignment Operators

Expression	Meaning
x=y	x=y
x+=y	x=x+y
x-=y	x=x-y
x*=y	x=x*y
x/=y	x=x/y
x%=y	x=x%y

Increment and Decrement Operators

Operator	Meaning
a++ (post increment)	Use the current value of 'a' in the expression which 'a' resides and then increment 'a' by 1.
++a (pre increment)	Increment 'a' by 1, then use the new value of 'a' in the expression which 'a' resides.
a-- (post decrement)	Use the current value of 'a' in the expression which 'a' resides and then decrement 'a' by 1.
--a (pre decrement)	decrement 'a' by 1, then use the new value of 'a' in the expression which 'a' resides.

Comparison Operators

Operator	Meaning	Example
==	Equal	a==10
!=	Not equal	a!=10
<	Less than	a<10
>	Greater than	a>10
<=	Less than or equal	a<=10
>=	Greater than or equal	a>=10

Logical Operators

Operator	Meaning	Example
AND (&&)	Returns true only if both sides in an expression are true. If any side fails, the operator returns false.	a>1 && a<10
OR (||)	Returns true atleast any side in an expression is true. If both sides fail, the operator returns false.	a>1 || a<10
XOR (^)	Returns true only one side in an expression is true. If both sides are true or both sides are false, the operator returns false.	a>1 ^ a<10
NOT (!)	The value of NOT is the negation of the expression	!(a<10)

Bitwise Operators

Operator	Meaning
&	Bitwise AND
|	Bitwise OR
^	Bitwise XOR
<<	Left shift
>>	Right shift
~	Bitwise complement

Access Modifiers

Default Access Modifier

variable or method is available to any other class in the same package

classes which are outside the package are not allowed to access the ariables or methods declared without specifying the access control modifier

Public Access Modifier

makes a method or a variable completely available to any other class

Private Access Modifier

completely hide a method or a variable from being accessed by any other class

Neither private variables nor private methods are inherited by sub classes.

Protected Access Modifier

Subclasses of a class

Other classes in the same package

Ternary (Conditional) Operator

```
class Constructs
{
        public static void main(String arg[])
        {
                int avg=75;
                char result;

                result=avg>50 ? 'A' : 'F' ;
                System.out.println(result);
        }
}
```

Arrays, Strings

Declaring One Dimensional Arrays

int numbers []

float [] marks

Note that the square brackets are used to specify as an array and it can be used either with the array name or the data type

Strings

Command Line Arguments

The JAVA program stores the command line arguments in the array of Strings which is passed to the man method in the program

Object Oriented Properties

Encapsulation

Encapsulation is the ability of an object to be a container for related properties and behaviour

Data hiding is the ability of objects to shield variables from external access. Data hiding is achieved by using Access modifiers

Inheritance

Specialization

Starting with something general and creating a refined version of it, is known as Specialization

Generalization

Taking many objects and finding their common basis is known as Generalization.

Syntax for Inheritance

class *class_name* extends *superclass_name*

Polymorphism

refers to the fact that a single operation can have different behaviors in different objects

Overloading Methods

Methods have the Same method name but method signature and the coding are different in sub classes

Overriding Methods

Methods have the same method signature only the coding is different

Polymorphism = Inheritance + Dynamic binding

Dynamic Binding

Late binding or Virtual binding

compiler is unable to resolve the procedure call and the binding is done at runtime

Static Binding

Early binding

compiler can resolve the binding at compile time

```
public class TestPolymorphism
{
     public static void main(String args[])
     {

          Person p;
          Employee e1=new Employee("Nimal",32,"Emp001");

          Student s1=new Student("Amila",22,"S/09/000");

          p=e1;
          p.display();
          p=s1;
          p.display();
     }
}
```

Dynamic binding

Name: Nimal
Age: 32
Emp No: Emp001

Name: Amila
Age: 22
Reg No: S/09/000

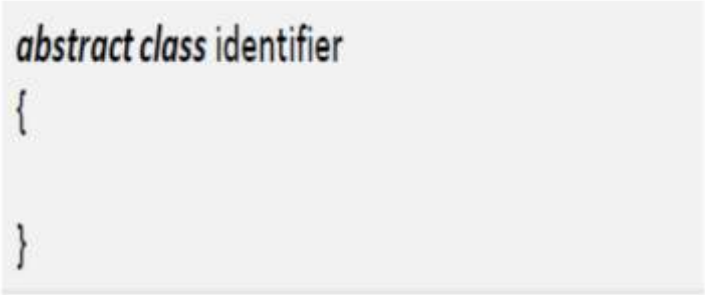
Abstraction

Abstraction is the act of representing essential features without giving background details or explanations

There are two ways to achieve abstraction

Abstract classes

cannot be instantiated



can be used as the super classes of Inheritance
They may OR may not contain abstract methods

Abstract Method
declared without an implementation
derived class should implement all the abstract methods in the base class

6. NOTE:
    a. If a class is defined as 'abstract', it may or may not contain any abstract methods.
    b. If a class does not contain any abstract method, then it can be defined as abstract only if the programmer needs to restrict the creation of objects of that class.
    c. If a class is not abstract, then it cannot hold any abstract method

        i. Interfaces
            1. named collection of method definitions (without implementations).
            2. There are no attributes defined in the interfaces. If the attributes are defined, they are implicitly 'static' and 'final'.
            3. class can implement any number of interfaces
            4. interfaces could be used to facilitate multiple inheritance with java