JAVA PROGRAMMING

Topic: OOPS CONCEPTS

SAVIYA VARGHESE
DEPT OF BCA
LITTLEFLOWER COLLEGE GURUVAYOOR

2020-21

MODULE 1:Introduction to OOPS

OBJECT ORIENTED PROGRAMMING(OOP):

- **Object-oriented programming** (**OOP**) is a programming paradigm based on the concept of "objects", which can contain data, in the form of fields (often known as *attributes* or *properties*), and code, in the form of procedures (often known as *methods*).

- OOP allows decomposition of a problem into a no. of entities called objects and then builds data and function around these objects.

Features of OOP

- Emphasis on data rather procedure.
- Programs are divided into units called “Objects”.
- Data structures are designed such that they characterize the objects
- Functions that operate on the data of an object are ties together in the data structure.
- Data is hidden and can’t be accessed by external function
- Objects may communicate with each other through function
- New functionality can be easily developed by creating objects and functions
- Follows bottom up approach in program design.

POP

- POP(procedure oriented programming) depends on procedures,which are sequence of procedures.
- Based on modular approach(larger pgms broken into small units –fns)
- Each procedure is a set of instructions that are executed one after another.
- Top down design
- Functional decomposition(systematically break the pbm into sub pbms)
- S/w maintenance can be difficult.
- Time consuming

POP Vs OOP

BASIS	POP	OOP
BASIC	Procedure/structure oriented	Object oriented
Approach	Top down	Bottom up
divided into	functions	objects
importance	Functions as well as seq.of actions to be done	Data
Access secifiers	Pop dosen't have any access specifier	Oop has access specifiers
Expansion	To add new data &fn is not so easy	Provides an easy way to add new data and fn
Data access	Accessed freely from fn to fn	Data can't move easily from fn to fn
Data hiding	It doesn't have any proper way for hiding data(less secure)	Provides data hiding(more secure)

overloading	Not possible	possible
inheritance	No provision of inheritance	Achieved in 3 modes(public,private &protected)
examples	C,VB,FORTRAN,PASCAL	C++,JAVA,VB.NET,C#.NET

Concepts of OOP

- Objects
- Classes
- Abstraction
- Encapsulation
- Inheritance
- Polymorphism
- Message passing

1.Objects

- S/w objects are the basic run time entities in an object riented s/m.
- It consists of state and behaviour .
- An object stores its state in fields or attributes(variables)
- Exposes its behaviour through methods(fns).
- Pgming pbm is analyzed in term of objects & nature of communication b/w them
- An object is an instance of a class.
- It takes up space in m/y and have an associated address.
- When a pgm is executed ,the objects interact by sending msgs to one anoher

Essential features objects

- State
- Behaviour
- Identity

State:

- ➢ State defines the object whereas behaviour defines what the object does.
- ➢ State consists of a set of data fields with their current values.(attributes/properties)
- ➢ An object’s State is defined by the attributes& by the values these have

- An attribute is a feature of an object ,which distinguish fro other kinds of objects.
- It is static(it can’t be removed/altered).

BEHAVIOUR

Behaviour is how an object acts and reacts in terms of its state changes and msg passing.

Operations on objects

1.Modifier- alter the state

2.Selector- access the state(doesn’t alter)

3.Constructor-creates an object &initializes its state

4.Destructor- destroys an object(free its m/y)

Identity

- Since object occupy a space in m/y so they have unique address in m/y. this become the identity of an object.

2.Classes

- A class is an abstract description of a set of objects
- It contains the descriptions of all the behaviours of the objects
- The individual behaviours of a class are called its methods
- A class is a template for the creation of a particular type of object.

  ie,We can use a class to create objects of the type described by the class.
- A class implementation consists of 2 parts-interface & implementation.
- Interface is outside view of the class(visible to everbody).
- Implementation is the actual code that implements the behaviour of the class

classes

- Class concept is necessary to class consumers who use the data types in their applns &keeps everything else hidden.(hiding the implementation reduces pgm bugs)

Abstraction

- Abstraction is the process of hiding the details and exposing only the essential features of a particular concept or object
- 2 types-DATA abstraction &FUNCTIONAL abstraction
- Data abstraction refers to the data that can be used
- Functional abstraction refers to a fn that can be used.
- In data abstraction ,access to the data takes is provided through a specific set of operations.
- In functional abstraction, access to the fn is provided through a specific interface

Abstraction

- Java takes the data abstraction one step further by supporting classes
- A class is a set of objects that share a common structure and a common behaviour
- Thae aim of the object oriented design approach is to build a set of abstractions that model the pbm domain.

Advantages:

1. Flexibility in approach:-by hiding data or abstracting details that are needed for presentation , the pgmer achieves greater flexibility in approach.

Abstraction

- 2.Enhanced security:- Abstraction gives access to data or details that are needed by users and hide the implementation details , giving enhanced security to appln.
- 3.Easier replacement:-Possible to replacecode without recompilation. It makes the process easier and saves time for users
- 4.modular approach:It helps users to divide an appln into modules &test each of them separately.Then all modules are integrated and ultimately tested together.It makes the appln devt easier.

Encapsulation

- It focuses upon the implementation that give rise to object behaviour.
- It is achieved through information hiding
- In java encapsulation is implemented through the concept of classes .
- Using the method of encapsulation,an object of a class cannot directly access the data,which is only accessible through the fns present inside the class

Difference b/w abstraction & encapsulation

abstraction	encapsulation
Solves the pbm in the design level	Solves the pbm in the implementation level
Used for hiding the unwanted data &giving relevant data	Hiding the code and data into a single unit to protect the data from outside world
Abstraction lets you focus on what the object does instead of how it does it	Hiding the internal details of how an object does something.
Outer layout ,used in terms of design	Inner layout ,used in terms of implementation

Inheritance

- In oop,inheritance is a way to form new classes using classes that have already been defined.

- It is the process of by which the objects of one class acquire the properties of objects of another class.

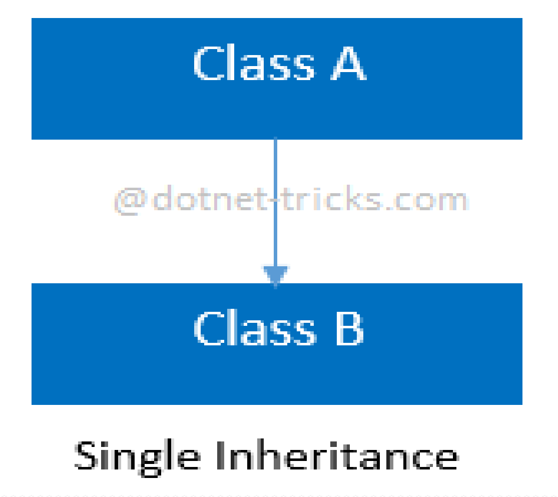
- The new classes-sub-classes(derived classes) inherit attributes and behaviour of the pre-existing classes(base classes).

Different types of inheritance

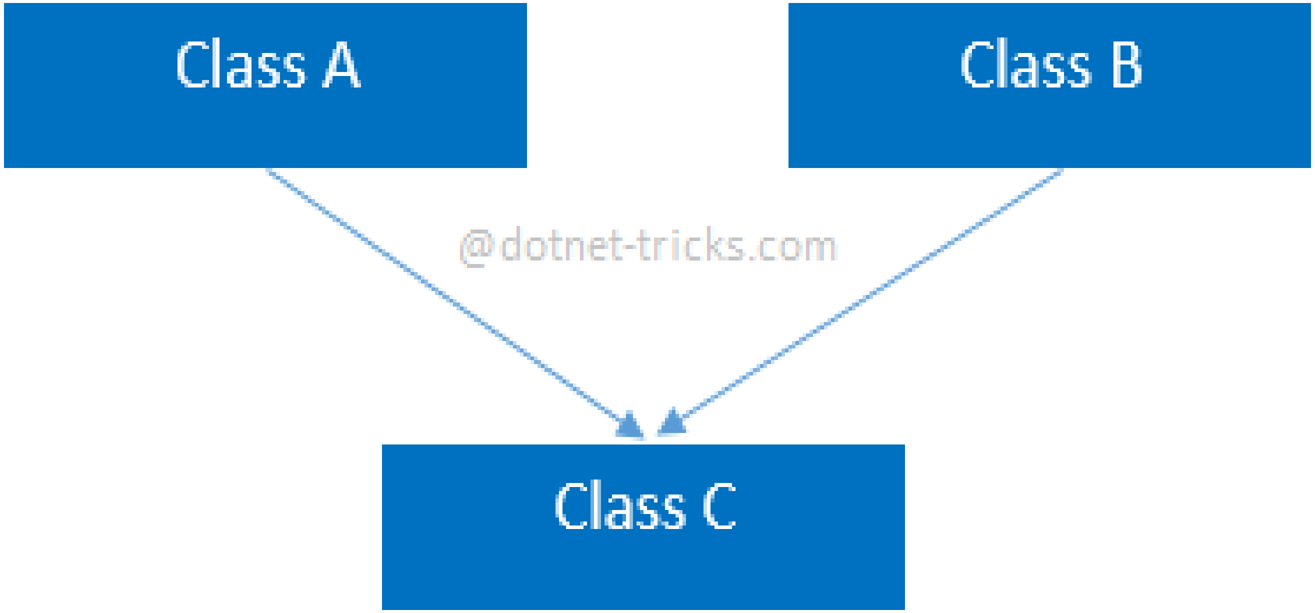
- Single
- Multiple
- Multilevel
- Hierarchical
- hybrid

Single Inheritance

- In this inheritance, a derived class is created from a single base class.
- In the given example, Class A is the parent class and Class B is the child class since Class B inherits the features and behavior of the parent class A



Single Inheritance



Class A

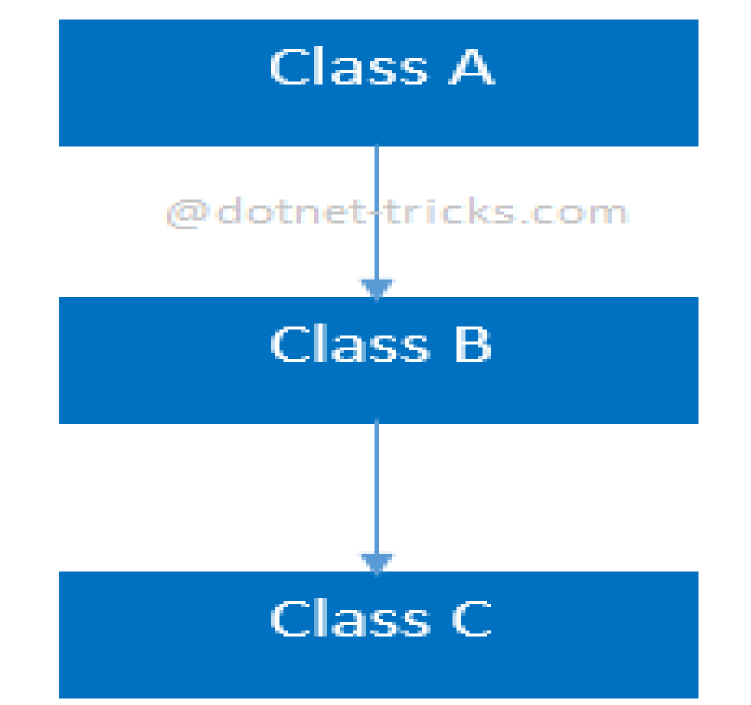
Class B

Class C

Multiple Inheritance

Multiple Inheritance

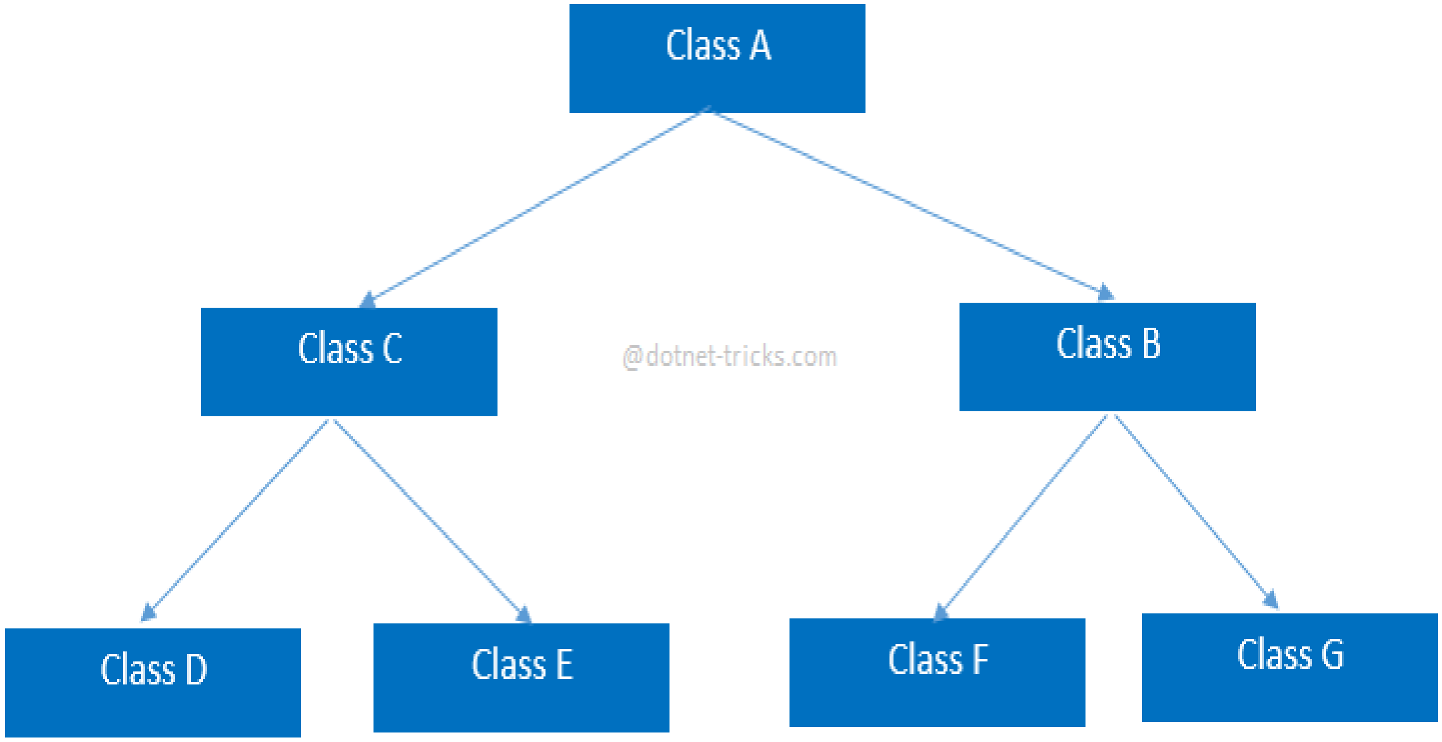
- In this inheritance, a derived class is created from more than one base class.
- This inheritance is not supported by .NET Languages like C#, F# etc. and Java Language.
- In the given example, class c inherits the properties and behavior of class B and class A at same level. So, here A and Class B both are the parent classes for Class C.



Multi-Level Inheritance

Multilevel Inheritance

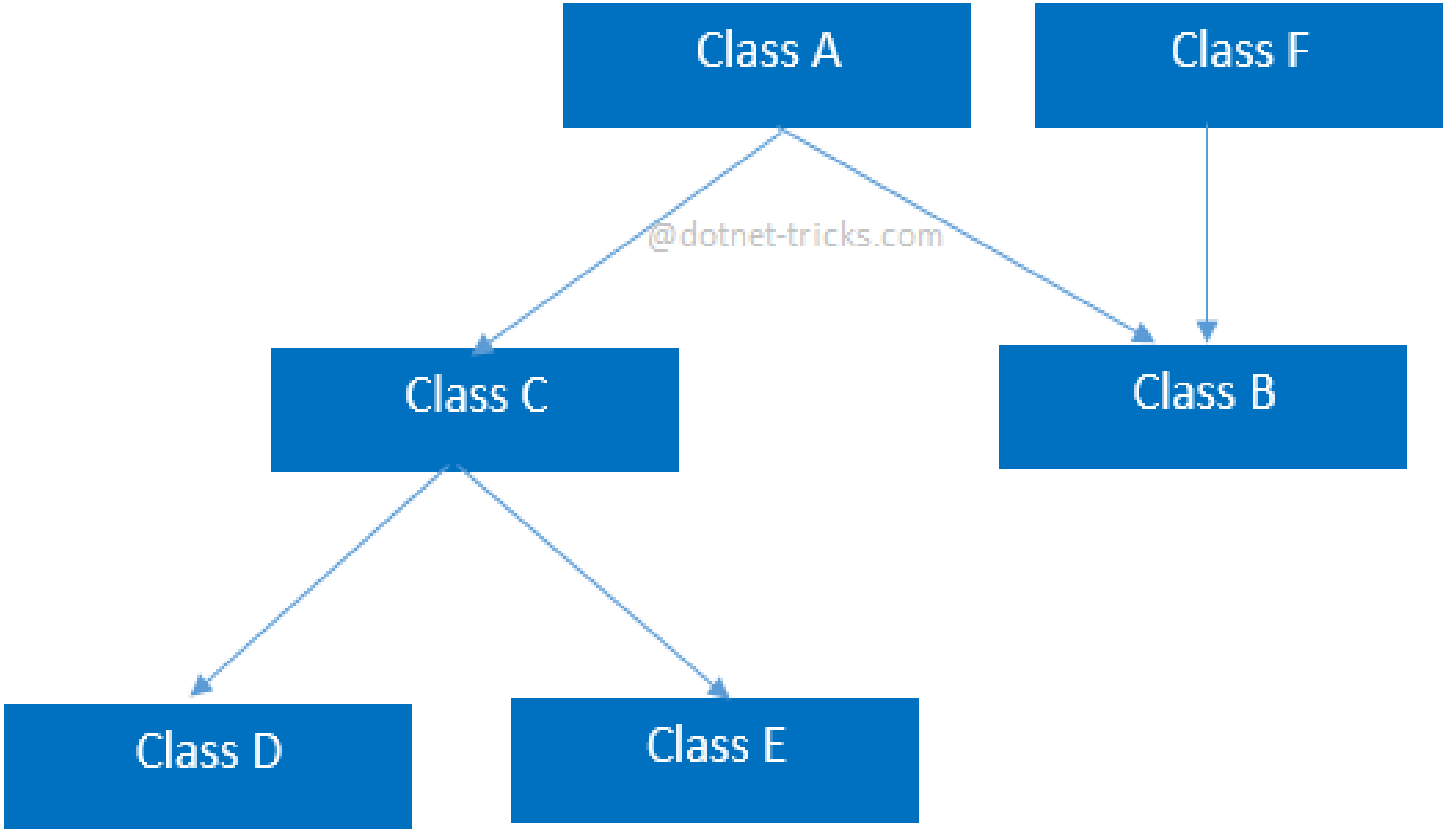
- In this inheritance, a derived class is created from another derived class.
- In the given example, class c inherits the properties and behavior of class B and class B inherits the properties and behavior of class B. So, here A is the parent class of B and class B is the parent class of C. So, here class C implicitly inherits the properties and behavior of class A along with Class B i.e there is a multilevel of inheritance.



Hierarchical Inheritance

Hierarchical Inheritance

- In this inheritance, more than one derived classes are created from a single base class and futher child classes act as parent classes for more than one child classes.
- In the given example, class A has two childs class B and class D. Further, class B and class C both are having two childs - class D and E; class F and G respectively.



Hybrid Inheritance – (a combination of Hierarchical and multiple)

Hybrid Inheritance

- This is combination of more than one inheritance. Hence, it may be a combination of Multilevel and Multiple inheritance

polymorphism

- Polymorphism means the ability to take more than one form.
- It is extensively used in implementing inheritance
- There are 2 types of polymorphism in Java

➢ Compile time polymorphism

➢ Runtime time polymorphism

<u>Compile time polymorphism</u>

- Static polymorphism
- Polymorphism that is resolved during compiler time is known as compile time polymorphism

Runtime polymorphism

- Dynamic polymorphism
- It is a process in which a call to an overridden method is resolved at runtime.

Message passing

- The term msg passing is the dynamic process of asking an object to perform a specific action.
- There are 3 identifiable parts to any message-passing expression.

➢ receiver

➢ msg selector

➢ arguments