INHERITANCE

When object acquires the properties and behaviour of parent object, it is known as inheritance. It provides code reusability.

POLYMORPHISM

If one task is performed in different way

Method overloading and overriding to achieve polymorphism

ABSTRACTION

Hiding internal details and showing only functionality

We use abstract class and interface to achieve abstraction

ENCAPSULATION

Binding code and data together into a single unit are known as encapsulation.

COUPLING

It refers to knowledge or information or dependency of another class.

If a class has info about another class then there is strong coupling.

We use public, private and protected to display visibility of class.

Use interface for weaker coupling.

ASSOCIATION

It represents the relationship between objects.

: one to one

: one to many

: many to one

: many to many

Advantages of oops over procedure programming

1. Dev and maintainance easier
2. Provides data hiding

Object:- an entity that has state and behaviour

An object is an instance of class.

Class is an template or blueprint from which objects are created.

Class is a group of objects which have common properties

It contains:-

Fields

Methods

Constructors

Blocks

Nested class and interface

Instance variable :- variable inside class but outside method. It doesn’t get memory on compile time but only during run time. That Is why it is called instance variable.

Methods :- code reusability and optimization

New:- it is used to allocate memory in runtime.

Class Student{

Int id; // instance variable

String name; // instance variable

Public static void main(String[] args){

Student s1 = new Student(); // object creation

System.out.println(s1.id);

}

}

3 ways to initialize object

By reference, by method, by constructor

An object which has no reference is called anonymous object.

Constructor is called when instance of class is created.

At the time of calling constructor, memory for object is allocated in memory.

If no constructor is provided it calls default constructor

Rules for Constructor Creation

1) Name should be same as class

2) must not have explicit return type

3) cannot be abstract, static, final and synchronized.

There are 2 types of constructor

Default constructor (no-arg constructor)

Parameterized Constructor

Class Animal{

Animal(){ // default constructor

System.out.println(“Animal object created”);

}

}

Class Car{

String name;

Int model;

Car(String n, int m){ // parameterized Constructor

Name= n;

Model = m;

}

}

Car obj = new Car(“maruti”,2015);

Obj.model; // 2015

Constructor overloading is a technique of having more than 1 constructor with different parameterised list.

Copy object

Class car{

String name;

Int model;

Car(Car obj){

Name = obj.name;

Model = ibj.model;

}

}

Car obj1 = new Car(“ford”,2019);

Car obj2 = new Car(obj1);

STATIC KEYWORD

Static keyword in java is used for memory management mainly.

We can apply it with variables,methods,blocks and nested class.

Variable

The static variable can be used to refer to common property of all objects

The static variable gets memory only once in the class area at the time of class loading.

Java Static property is shared across all objects.

Advantages of Static Variable

1. It makes program memory efficient.

METHOD

It can be invoked without the need for creating an instance of class

Class Student{

Static void printCollege(){

System.out.println(“Chitkara University”);

}

}

To access use className.method() directly without creating instance

Student.printCollege();

It cannot use non static data members or call non static methods directly

This and super cannot be used in static context.

Java main class is static so that JVM doesn’t have to create an object to call it and unnecessary memory is not used.

This Keyword

In java this is a reference variable that refers to the current object.

1. It can be used to access current class instance variable
2. It is used to invoke current class methods
3. This() to invoke current class constructor
4. Can be used to return current class instance