Ploymorphism -- which literally means many shapes -- is the ability of objects to belong to different types.

Dynamic binding is not a type of polymorphism; rather dynamic binding is how polymorphism is "implemented" or how it works. And dynamic binding and runtime binding are the same thing. "Dynamic Binding" is the more academic or computer science based name for it. And "run time binding" is a more every day term for it. And as if that wasn't enough, it can also be called "late binding".

**Static binding/Compile-Time binding/Early binding/Method overloading.(in same class)**

class Calculation {

void sum(int a,int b){System.out.println(a+b);}

void sum(int a,int b,int c){System.out.println(a+b+c);}

public static void main(String args[]) {

Calculation obj=new Calculation();

obj.sum(10,10,10); // 30

obj.sum(20,20); //40

}

}

**Dynamic binding/Run-Time binding/Late binding/Method overriding.(in different classes)**

class Animal {

public void move(){

System.out.println("Animals can move");

}

}

class Dog extends Animal {

public void move() {

System.out.println("Dogs can walk and run");

}

}

public class TestDog {

public static void main(String args[]) {

Animal a = new Animal(); // Animal reference and object

Animal b = new Dog(); // Animal reference but Dog object

a.move();//output: Animals can move

b.move();//output:Dogs can walk and run

}

}