**Polymorphism in Java**

Poly means Many

Morphism means form

Q.1 What is Polymorphism?

* One functionality with different form
* If we develop any feature such that it can make more than one form depending on the situation is called polymorphism
* One function with different behavior i.e also called as Polymorphism

Q.1 How we are going to achieve polymorphism in java or Types of polymorphism in java

1. Compile time

2. run time

1. Compile time Polymorphism is also known as static binding or early binding

2. run time Polymorphism is also known as dynamic binding or late binding

1**. Compile time Polymorphism**: we will achieve compile time polymorphism in three ways

a. Method overloading

b. constructor overloading

c. operator overloading

**2. Runtime Polymorphism**

Runtime polymorphism we will achieve by using method overriding only

We are declare method

public void test(int a){

// Business logic

}

Public static void test(int a){

// Business logic

}

Void test(int a): method signature

void : return time of method

public: Access modifier in java

**a. Compile time polymorphism or static binding or early binding**

1. **Method overloading**: here method name must be same but then we will differentiate the method based on method signature or type of the argument within the method or based on the argument within the method

**Program1:** method signature must be different

**package** pack1;

**public** **class** A {

// while overloading the method method signature must be different

**void** test() {

System.***out***.println("ins test()method");

}

**void** test(**int** a) {

System.***out***.println("ins test(int a)method");

}

**void** test(**char** ch) {

System.***out***.println("ins test(char ch)method");

}

**public** **static** **void** main(String[] args) {

A a = **new** A();

a. test();

a.test('A');

a. test(100);

}

}

// output

//ins test()method

//ins test(char ch)method

//ins test(int a)method

**Program2:** Here you identify method signature must be different or not?

Ans is yes we are writing method signature different in sequence.. because

1. in 1st method we are writing int a first and char ch second

2. in 2nd method int a second and char ch 1st in bracket

**package** pack1;

**public** **class** B {

**void** test( **int** a, **char** ch) {

System.***out***.println("ins test(int a,char ch)");

}

**void** test(**char** ch, **int** a) {

System.***out***.println("ins test(char ch, int a)");

}

**public** **static** **void** main(String[] args) {

B b =**new** B();

b.test(10,'A');

b.test('B', 20);

}

}

// Output

//ins test(int a,char ch)

//ins test(char ch, int a)