



LECTURE

POLYMORPHISM

Dr. Rubeena DOOMUN
Open University of Mauritius

METHOD OVERLOADING

- **Method Overloading** is a feature that allows a class to have more than one **method** having the same name, but their argument lists are different.

```
public 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 cal = new Calculation();  
        cal.sum(20,30,60);  
        cal.sum(20,20);  
    }  
}
```

THREE WAYS TO OVERLOAD A METHOD

- 1. Number of parameters
- 2. Data type of parameters
- 3. Sequence of Data type of parameters

- **Method overloading** is an example of Static Polymorphism.
- Static Polymorphism is also known as compile time binding or early binding.
- Static binding happens at compile time.
Method overloading is an example of static binding where binding of method call to its definition happens at Compile time.

METHOD OVERRIDING

- **Method overriding**, in object-oriented programming, is a language feature that allows a subclass or child class to provide a specific implementation of a **method** that is already provided by one of its superclasses or parent classes.

RULES FOR METHOD OVERRIDING

- Method must have same name as in the parent class.
- Method must have same parameter as in the parent class.
- Must be IS-A relationship (inheritance).

- Method overriding is used to provide specific implementation of a method that is already provided by its super class.
- Method overriding is used for runtime polymorphism.

Overloading	Overriding
Method overloading is used <i>to increase the readability</i> of the program.	Method overriding is used <i>to provide the specific implementation</i> of the method that is already provided by its super class.
Method overloading is performed <i>within class</i> .	Method overriding occurs <i>in two classes</i> that have IS-A (inheritance) relationship.
In case of method overloading, <i>parameter must be different</i> .	In case of method overriding, <i>parameter must be same</i> .
Method overloading is the example of <i>compile time polymorphism</i> .	Method overriding is the example of <i>run time polymorphism</i> .
In java, method overloading can't be performed by changing return type of the method only. <i>Return type can be same or different</i> in method overloading. But you must have to change the parameter.	<i>Return type must be same or covariant</i> in method overriding.

REFERENCES

- https://www.tutorialspoint.com/java/java_interfaces.htm
- https://www.tutorialspoint.com/javaexamples/method_overloading.htm
- <https://www.javatpoint.com/method-overriding-in-java>