CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

INSTITUTE : UIE

DEPARTMENT : CSE

Bachelor of Engineering (Computer Science & Engineering)

PROJECT BASED LEARNING IN JAVA

(20CST-319/20ITT-319)

TOPIC OF PRESENTATION:

Difference between method overloading and method overriding.

Lecture Objectives

In this lecture, we will discuss:

Difference between method overloading and method overriding.

Method Overriding

If subclass (child class) has the same method as declared in the parent class, it is known as **method overriding in Java**.

- Method with the same name, arguments and return type in both the parent and the child class

- If an overridden method is called from subclass object , it will always refer to the version defined by the subclass

- **Need to Override**– If a sub class **needs** have a different method implementation from that of a super class

Usage of Java Method Overriding

- Method overriding is used to provide the specific implementation of a method which is already provided by its superclass.
- Method overriding is used for runtime polymorphism

Rules for Java Method Overriding

- The method must have the same name as in the parent class
- The method must have the same parameter as in the parent class.
- There must be an IS-A relationship (inheritance).

Method Overriding

```
public class Shape {

    void draw()
    {
        System.out.println("Drawing Shape");
    }
    void erase()
    {
        System.out.println("erasing Shape");
    }
```

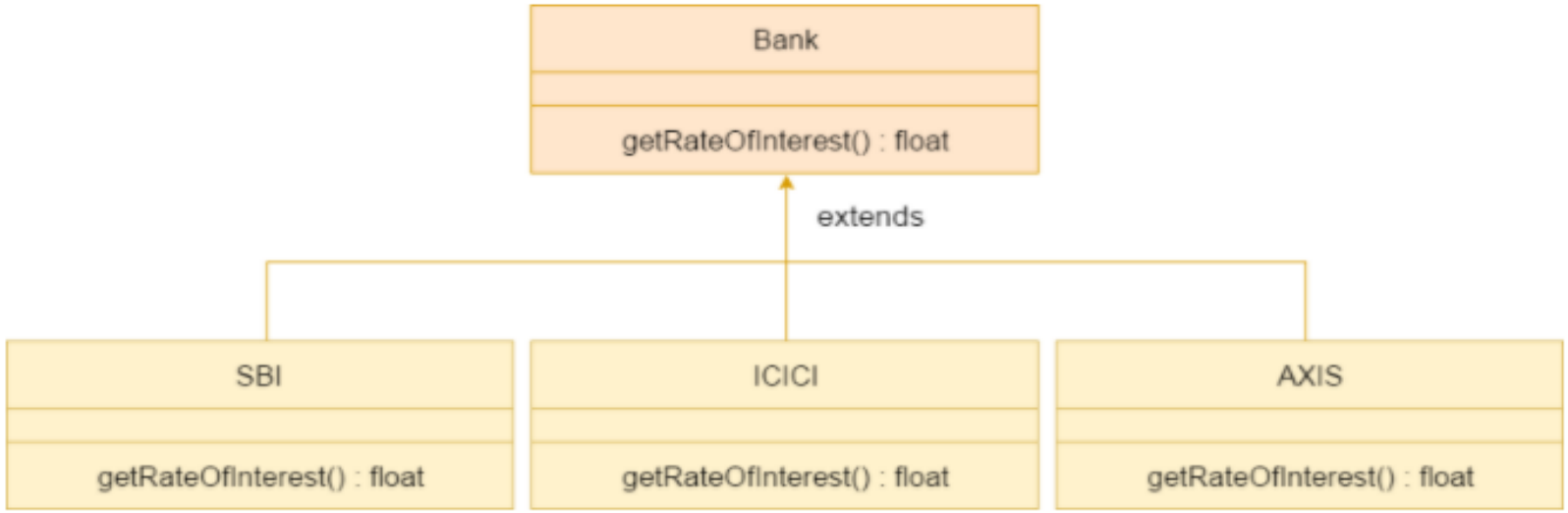
Draw() and erase() in Base Class

```
class Square extends Shape
{
    void draw()
    {
        System.out.println("Drawing Square");
    }
    void erase()
    {
        System.out.println("erasing Square");
    }
}
```

Draw() and erase() in Derived Class

Method Overriding

Another Example:



Can we override static method?

No, a static method cannot be overridden. It can be proved by runtime polymorphism.

Why can we not override static method?

It is because the static method is bound with class whereas instance method is bound with an object. Static belongs to the class area, and an instance belongs to the heap area.

Can we override java main method?

No, because the main is a static method.

Dynamic Method Dispatch or Runtime Polymorphism

Basis of one of Java's most powerful concepts: dynamic method dispatch

- Dynamic method dispatch occurs when the Java language resolves a call at runtime

•Runtime polymorphism is possible in java using

  - superclass reference variables
  - overridden methods

Method Overloading

Method Overloading in Java

- If a class has multiple methods having same name but different in parameters, it is known as **Method Overloading**.
- If we have to perform only one operation, having same name of the methods increases the readability of the program.

Allows to define the method with the same name but either return type or arguments list should be different

```
Class BasicCalculator{
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);
}
```

Advantage of method overloading

Method overloading *increases the readability of the program*.

Different ways to overload the method

There are two ways to overload the method in java

- By changing number of arguments
- By changing the data type

Q) Why Method Overloading is not possible by changing the return type of method only?

In java, method overloading is not possible by changing the return type of the method only because of ambiguity. Let's see how ambiguity may occur:

```
class Adder{
static int add(int a,int b){return a+b;}
static double add(int a,int b){return a+b;}
}
class TestOverloading3{
public static void main(String[] args){
System.out.println(Adder.add(11,11));//ambiguity
}}
```

Output:

Compile Time Error: method add(int,int) is already defined in class Adder

System.out.println(Adder.add(11,11)); //Here, how can java determine which sum() method should be called?

No.	Method Overloading	Method Overriding
1)	Method overloading is used *to increase the readability* of the program.	Method overriding is used *to provide the specific implementation* of the method that is already provided by its super class.
2)	Method overloading is performed *within class*.	Method overriding occurs *in two classes* that have IS-A (inheritance) relationship.
3)	In case of method overloading, *parameter must be different*.	In case of method overriding, *parameter must be same*.
4)	Method overloading is the example of *compile time polymorphism*.	Method overriding is the example of *run time polymorphism*.
5)	In java, method overloading can't be performed by changing return type of the method only. *Return type can be same or different* in method overloading. But you must have to change the parameter.	*Return type must be same or covariant* in method overriding.

QUIZ:

1. What is the process of defining two or more methods within same class that have same name but different parameters declaration?
   a) method overloading
   b) method overriding
   c) method hiding
   d) none of the mentioned
2. What is the process of defining a method in terms of itself, that is a method that calls itself?
   a) Polymorphism
   b) Abstraction
   c) Encapsulation
   d) Recursion

Summary:

In this session, you were able to :

- Learn about Difference between method overloading and method overriding.

References:

Books:

1. Balaguruswamy, *Java.*
2. A Primer, E.Balaguruswamy, *Programming with Java,* Tata McGraw Hill Companies
3. John P. Flynt Thomson, *Java Programming.*

Video Lectures :

https://www.youtube.com/watch?v=RaAp-f2Np1U
https://youtu.be/9OXXbQBesHg

Reference Links:

https://www.javatpoint.com/method-overloading-in-java
https://www.javatpoint.com/method-overriding-in-java
https://www.javatpoint.com/method-overloading-vs-method-overriding-in-java
https://www.geeksforgeeks.org/overloading-in-java/
https://www.journaldev.com/16807/method-overloading-in-java

THANK YOU