

Object Oriented Programming using Java

Prepared By:
Suyel, PhD
Assistant Professor
Dept. of CSE, NIT Patna



Outline

1. Method Overloading
2. Examples of Method Overloading
3. Method Overloading with Type Promotion

Method Overloading

- ❑ If a class has multiple methods having same name, but different in parameters, it is called **Method Overloading**.
- ❑ Here, different number of parameters, different types of parameters or both are considered.
- ❑ Java supports method overloading in the same class.
- ❑ Overloaded methods may have the same or different return types, but they must differ in parameters.
- ❑ Method overloading is not possible by changing the return type of the method only because of ambiguity. Compiler cannot be able to determine which method should be called.

Method Overloading (Cont...)

- ❑ Method overloading is one of the ways through which java supports polymorphism (i.e., static polymorphism).
- ❑ We can have any number of main methods in a class by method overloading. However, JVM only calls main() method, which receives string array as arguments.
- ❑ Example:

```
void hello() { ... }
```

```
void hello(int x) { ... }
```

```
float hello(double x) { ... }
```

```
float hello(int x, float y) { ... }
```

Method Overloading (Cont...)

- ❑ There are two ways to overload the method in Java:
 - ❖ By changing the data type
 - ❖ By changing number of arguments
- ❑ **Advantages of method overloading:**
 - ❖ Method overloading increases the readability of the program.
 - ❖ It provides flexibility to programmers, so that we can call the same method for different types of data.
 - ❖ It reduces the execution time because the binding is done in compilation time itself.
 - ❖ Method overloading minimizes the complexity of the code.
 - ❖ We can use the code again and again, which saves memory.

Examples of Method Overloading

```
class Method1
{
    void sum(int a, int b)
    {
        System.out.println("sum: " + (a+b)) ;
    }

    void sum(float a, float b)
    {
        System.out.println("sum: " + (a+b));
    }

    public static void main (String args[])
    {
        Method1 obj = new Method1();
        obj.sum(5, 10);
        obj.sum(20.5f, 40.3f);
    }
}
```

Examples of Method Overloading (Cont...)

❑ Output

Sum: 15

Sum: 60.8

Examples of Method Overloading (Cont...)

```
class Method2
{
    void disp(char c)
    {
        System.out.println(c);
    }

    void disp(int c)
    {
        System.out.println(c);
    }

    public static void main(String args[])
    {
        Method2 obj = new Method2();
        obj.disp('A');
        obj.disp(10);
    }
}
```


Examples of Method Overloading (Cont...)

❑ Output

A

10

Examples of Method Overloading (Cont...)

```
class Method3
{
    static int sum(int a, int b)
    {
        return a+b;
    }

    static double sum(double a, double b)
    {
        return a+b;
    }

    public static void main(String args[])
    {
        System.out.println("Sum: " + sum(5, 10));
        System.out.println("Sum: " + sum(20.5, 40.3));
    }
}
```

Examples of Method Overloading (Cont...)

❑ Output

Sum: 15

Sum: 60.8

Examples of Method Overloading (Cont...)

Test.java

```
class Test
{
    private static void disp(int x)
    {
        System.out.println("It is an integer value");
    }

    private static void disp(String str)
    {
        System.out.println("It is a string");
    }
}
```

Method4.java

```
class Method4
{
    public static void main(String args[])
    {
        Test.disp(1);
        Test.disp("Hello! How are you?");
    }
}
```

Examples of Method Overloading (Cont...)

❑ Output

```
Method4.java:6: error: no suitable method found for disp(int)
    Test.disp(1);
        ^
    method Test.disp(int) is not applicable
      (disp(int) has private access in Test)
    method Test.disp(String) is not applicable
      (argument mismatch; int cannot be converted to String)
Method4.java:7: error: no suitable method found for disp(String)
    Test.disp("Hello How are you?");
        ^
    method Test.disp(String) is not applicable
      (disp(String) has private access in Test)
    method Test.disp(int) is not applicable
      (argument mismatch; String cannot be converted to int)
2 errors
```

Examples of Method Overloading (Cont...)

```
class Method5
{
    public int sum(int x, int y)
    {
        return (x + y);
    }

    public int sum(int x, int y, int z)
    {
        return (x + y + z);
    }

    public double sum(double x, double y)
    {
        return (x + y);
    }

    public static void main(String args[])
    {
        Method5 obj = new Method5();
        System.out.println(obj.sum(5, 10));
        System.out.println(obj.sum(5, 10, 20));
        System.out.println(obj.sum(20.5, 40.3));
    }
}
```

Examples of Method Overloading (Cont...)

❑ Output

15

35

60.8

Examples of Method Overloading (Cont...)

```
class Method6
{
    private static void disp(int x)
    {
        System.out.println("Argument: " + x);
    }

    private static void disp(int x, int y)
    {
        System.out.println("Argument 1: " + x + ", Argument 2: " + y);
    }

    public static void main(String args[])
    {
        disp(5);
        disp(5, 10);
    }
}
```


Examples of Method Overloading (Cont...)

❑ Output

Argument: 5

Argument 1: 5, Argument 2: 10

Examples of Method Overloading (Cont...)

Test1.java

```
class Test1
{
    public void disp(char c)
    {
        System.out.println("Character: " + c);
    }
    public void disp(char c, int x)
    {
        System.out.println("Character: " + c + ", Integer: " + x);
    }
}
```

Method7.java

```
class Method7
{
    public static void main(String args[])
    {
        Test1 obj = new Test1();
        obj.disp('A');
        obj.disp('A', 10);
    }
}
```

Examples of Method Overloading (Cont...)

❑ Output

Character: A

Character: A, Integer: 5

Examples of Method Overloading (Cont...)

```
class Method8
{
    int disp(int x)
    {
        return x;
    }

    double disp(int y)
    {
        return y;
    }

    public static void main(String args[])
    {
        Method8 obj = new Method8();
        System.out.println("Value of x: " + obj.disp(5));
        System.out.println("value of y: " + obj.disp(10.55));
    }
}
```

Examples of Method Overloading (Cont...)

❑ Output

```
Method8.java:8: error: method disp(int) is already defined in class Method8
    double disp(int y)
               ^
Method8.java:17: error: incompatible types: possible lossy conversion from double to int
    System.out.println("Value of y: " + obj.disp(10.55));
                                   ^
Note: Some messages have been simplified; recompile with -Xdiags:verbose to get full output
2 errors
```

Examples of Method Overloading (Cont...)

```
class Method9
{
    public static void main(int x)
    {
        System.out.println("Main method with Integer argument");
        System.out.println(x);
    }

    public static void main(char c)
    {
        System.out.println("Main method with Character argument");
        System.out.println(c);
    }

    public static void main(Double d)
    {
        System.out.println("Main method with Double argument");
        System.out.println(d);
    }

    public static void main(String args[])
    {
        System.out.println("This is the original Main method");
        Method9.main(10);
        Method9.main('A');
        Method9.main(20.5);
    }
}
```

Examples of Method Overloading (Cont...)

❑ Output

```
This is the Original Main method  
Main method with Integer argument  
10  
Main method with Character argument  
A  
Main method with Double argument  
20.5
```

Examples of Method Overloading (Cont...)

```
class Method10
{
    int ID;
    String name;

    Method10()
    {
        System.out.println("It is our Default Constructor");
    }

    Method10(int x, String str)
    {
        System.out.println("Now, it is our Parameterized Constructor");
        ID = x;
        name = str;
    }

    public static void main(String args[])
    {
        Method10 obj = new Method10();
        System.out.println("Values of Default Constructor: ");
        System.out.println(" Student ID: " + obj.ID + ", Student Name: " + obj.name);

        Method10 obj1 = new Method10(1010, "Shivam");
        System.out.println("Values of Parameterized Constructor: ");
        System.out.println(" Student ID: " + obj1.ID + ", Student Name: " + obj1.name);
    }
}
```


Examples of Method Overloading (Cont...)

❑ Output

```
It is our Default Constructor  
Values of Default Constructor:  
  Student ID: 0, Student Name: null  
Now, it is our Parameterized Constructor  
Values of Parameterized Constructor:  
  Student ID: 1010, Student Name: Shivam
```

Method Overloading with Type Promotion

- ❑ When a data type of smaller size is promoted to the data type of bigger size, it is called type promotion.
- ❑ For example: byte data type can be promoted to short, short data type can be promoted to int, long, double, etc.
- ❑ Java supports automatic type promotion, if no matching data type is found.
- ❑ If there are no matching type arguments in the method and each method promotes the similar type of arguments, then there is an ambiguity.

Method Overloading with Type Promotion (Cont...)

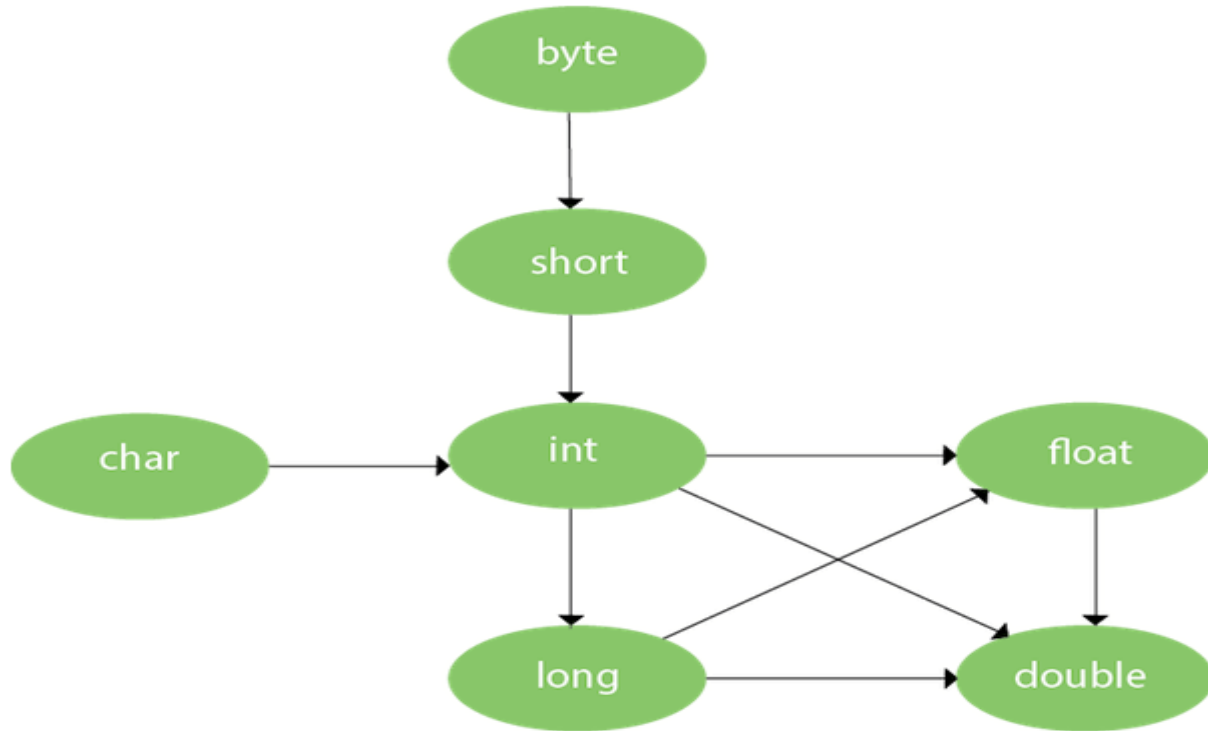


Fig. 1: Type promotion

Method Overloading with Type Promotion (Cont...)

```
class Method11
{
    void sum(int x, long y)
    {
        System.out.println("Sum: " + (x+y));
    }

    void sum(int x, int y, int z)
    {
        System.out.println("Sum: " + (x+y+z));
    }

    public static void main(String args[])
    {
        Method11 obj=new Method11();
        obj.sum(5, 10);
        obj.sum(5, 10, 20);
    }
}
```

Method Overloading with Type Promotion (Cont...)

❑ Output

Sum: 15

Sum: 35

Method Overloading with Type Promotion (Cont...)

```
class Method12
{
    public void show(int x)
    {
        System.out.println("Type Integer: " + x);
    }

    public void show(String str)
    {
        System.out.println("Type String: " + str);
    }

    public void show(byte b)
    {
        System.out.println("Type Byte: " + b);
    }

    public static void main(String[] args)
    {
        byte a = 10;
        Method12 obj = new Method12();
        obj.show(a);
        obj.show("How are you?");
        obj.show(20);
        obj.show('A');
        obj.show("A");
        //obj.show(20.5f);
    }
}
```

Method Overloading with Type Promotion (Cont...)

❑ Output

```
Type Byte: 10  
Type String: How are you?  
Type Integer: 20  
Type Integer: 65  
Type String: A
```

Method Overloading with Type Promotion (Cont...)

```
class Method13
{
    void sum(int x, long y)
    {
        System.out.println("Method 1 is invoked");
    }

    void sum(long x, int y)
    {
        System.out.println("Method 2 is invoked");
    }

    public static void main(String args[])
    {
        Method13 obj = new Method13();
        obj.sum(5, 10);
    }
}
```


Method Overloading with Type Promotion (Cont...)

❑ Output

```
Method13.java:17: error: reference to sum is ambiguous
    obj.sum(5, 10);
        ^
    both method sum(int,long) in Method13 and method sum(long,int) in Method13 match
1 error
```



**Slides are prepared from various sources,
such as Book, Internet Links and many
more.**