OVERLOADING
LECTURE 9

BITS-PILANI,K.K. BIRLA GOA CAMPUS

Prof. Anita Agrawal

Method Overloading???

- In Java, two or more methods can have the same name if ……
  - They differ in parameters:
    - different number of parameters,
    - different types of parameters,
    - Both of the above

Overloaded methods

Method overloading

Example:

```
void func()
{
   …
}
void func(int a)
{
    …
}
```

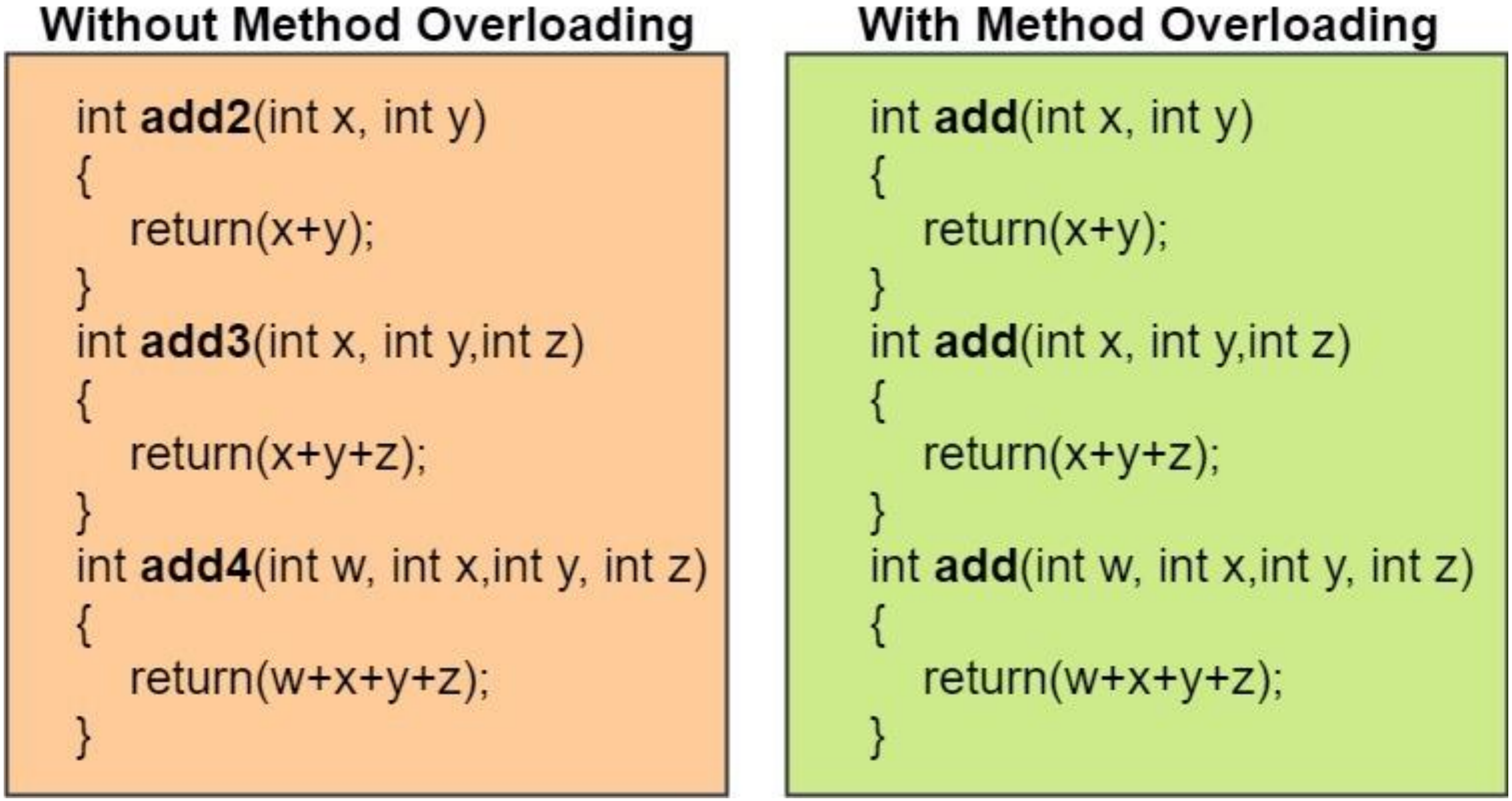
```
void func(double a)
 {
   …
 }
void func(int a, int b)
 {
   …
}
```

Example:

```
void func(int a)
{
     …
}
```

```
int func (int a)
 {
     …
 }
```

Overloaded methods may or may not have different return type, but they must differ in parameters they accept.



Without Method Overloading

```
int add2(int x, int y)
{
   return(x+y);
}
int add3(int x, int y,int z)
{
   return(x+y+z);
}
int add4(int w, int x,int y, int z)
{
   return(w+x+y+z);
}
```

With Method Overloading

```
int add(int x, int y)
{
   return(x+y);
}
int add(int x, int y,int z)
{
   return(x+y+z);
}
int add(int w, int x,int y, int z)
{
   return(w+x+y+z);
}
```

Method overloading, why???

- It improves the readability of the program.

- Only one method needs to be invoked

Different ways to perform method overloading…..

□ Overloading by changing number of arguments

□ Overloading by changing the type of arguments

Important Points

- Two or more methods can have same name inside the same class if they accept different arguments.

- Method overloading is achieved by either:
  - changing the number of arguments.
  - changing the datatype of arguments.

- Method overloading is not possible by changing the return type of methods.

Method Overloading…

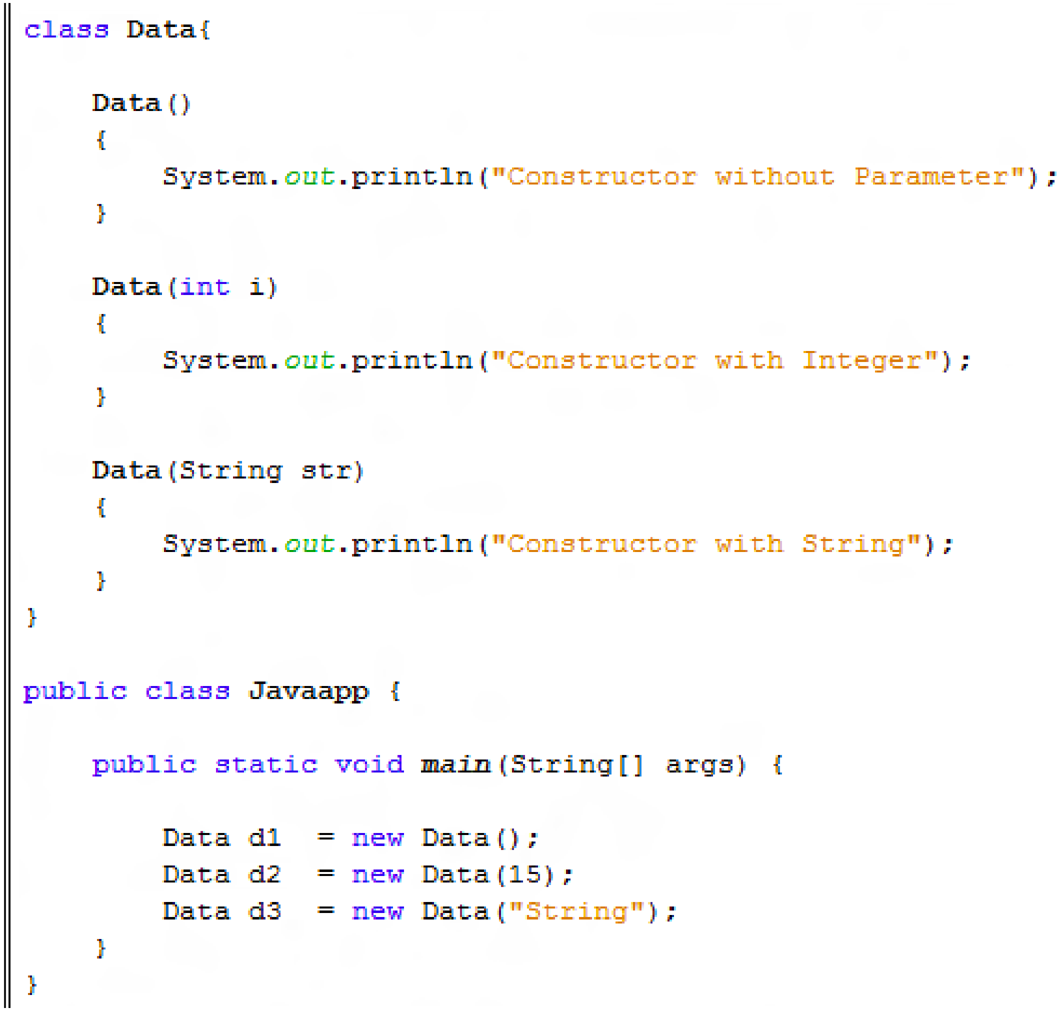
- When an overloaded method is called: –
  - Java first looks for an exact match between the arguments and method's parameters
    - Number and type is matched
    - If exact match method is not found
      - Automatic type conversion will take place-> Overload resolution

…over_convert.java

Constructor Overloading

- Like methods, constructors too can be overloaded in a class.

.....COL



```java
class Data{

    Data()
    {
        System.out.println("Constructor without Parameter");
    }

    Data(int i)
    {
        System.out.println("Constructor with Integer");
    }

    Data(String str)
    {
        System.out.println("Constructor with String");
    }
}

public class Javaapp {

    public static void main(String[] args) {

        Data d1  = new Data();
        Data d2  = new Data(15);
        Data d3  = new Data("String");
    }
}
```

Passing objects as parameters

- Just like primitive types, objects can also be used as parameters to methods.

- Constructors can also have Objects as parameters
- –Useful in duplicating objects

Argument Passing

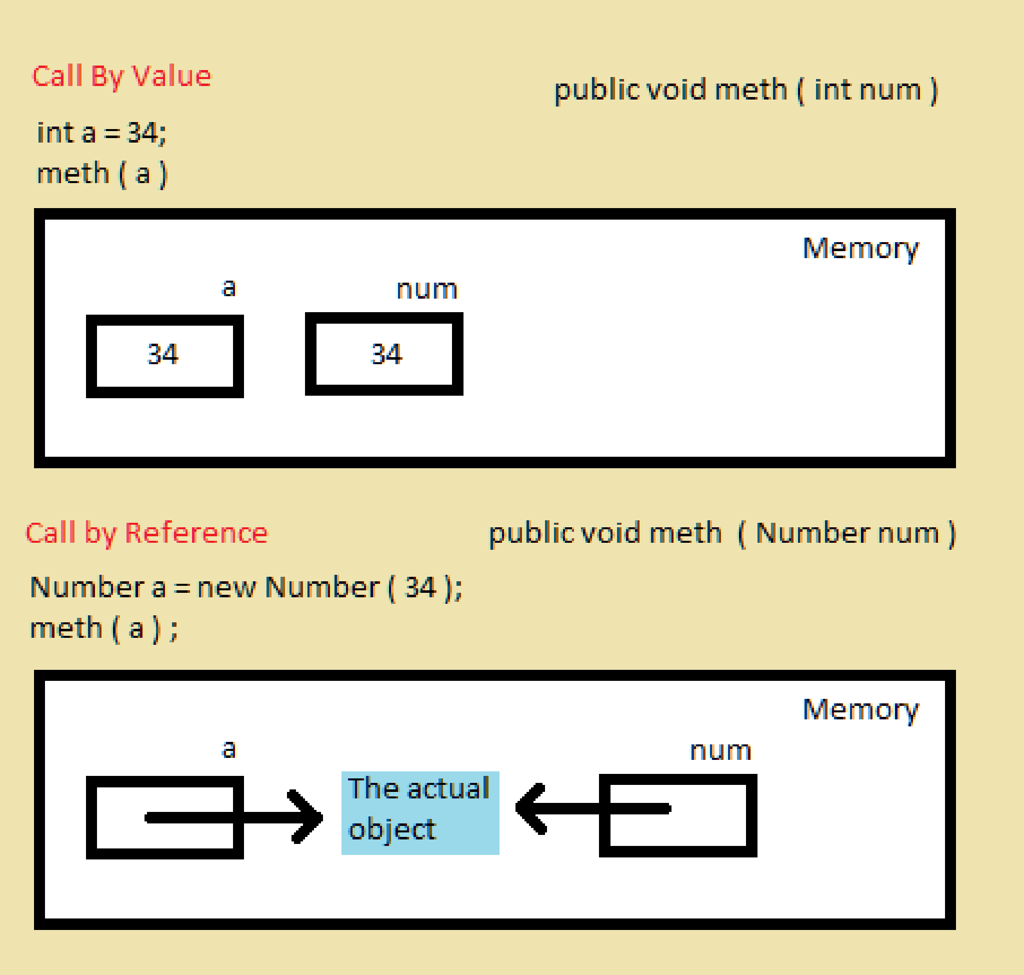
- Call-by-value: Value of argument passed to the parameter of subroutine

  Changes made to the parameter don’t affect the argument.

- Call-by-Reference: Reference to an argument is passed to the parameter of subroutine.

  Changes made to the parameter will affect the argument.

- In Java,
  - Primitive types: Call-by-value
  - Objects: call by reference



Call By Value

```
public void meth ( int num )
```

```
int a = 34;
meth ( a )
```

Call by Reference

```
public void meth  ( Number num )
```

```
Number a = new Number ( 34 );
meth ( a ) ;
```