



1. Parameters and arguments
2. Variable parameters

Method Parameters I

Parameters and arguments

1. Parameters and arguments

```
public void greeting(){  
    System.out.println("Hello world");  
}
```

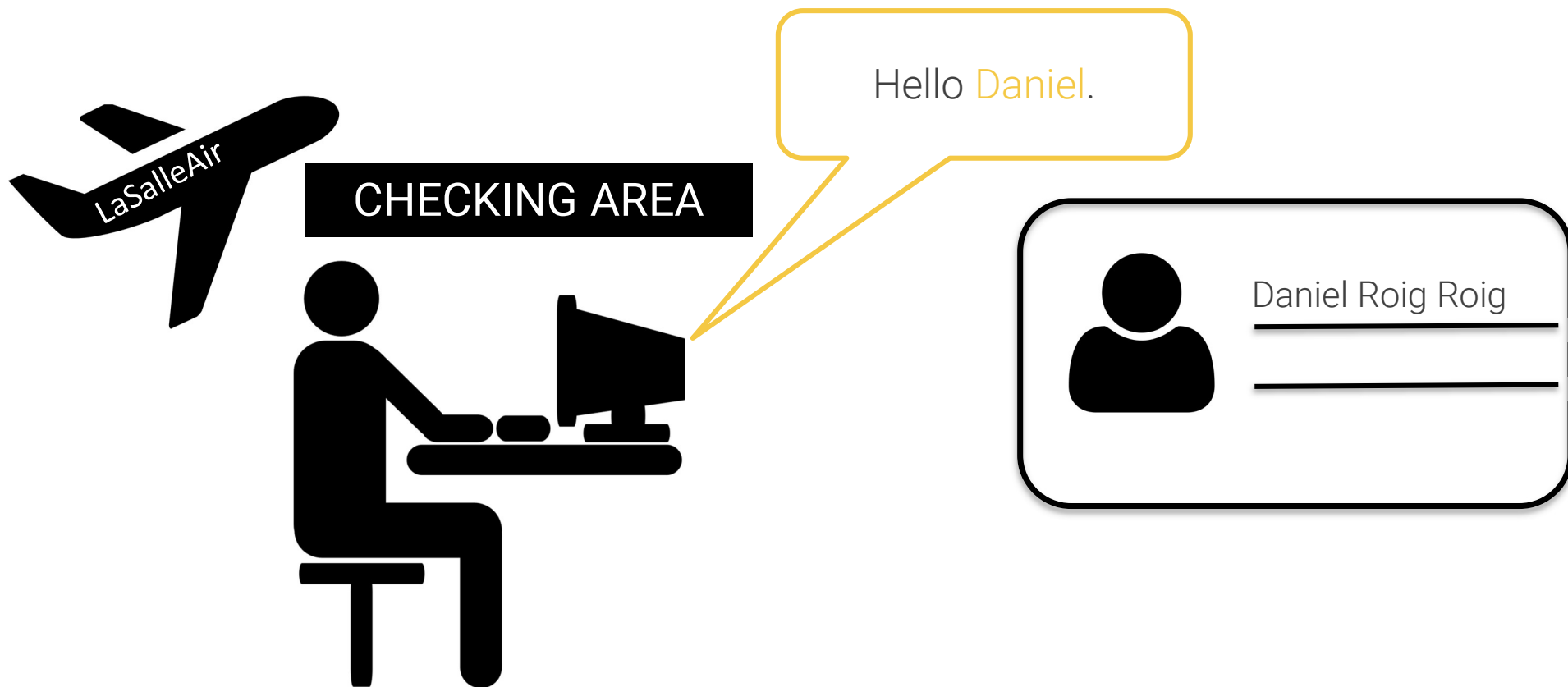
1. Parameters and arguments



1. Parameters and arguments

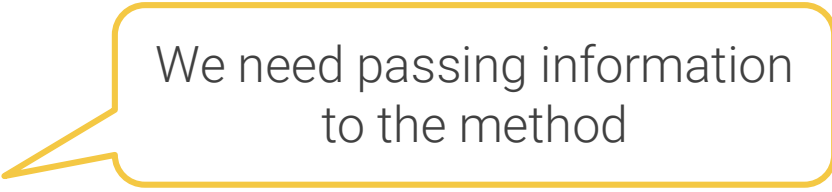
```
public void greeting(){  
    System.out.println("Hello Elena");  
}
```

1. Parameters and arguments



1. Parameters and arguments

```
public void greeting(){  
    System.out.println("Hello Daniel");  
}
```



We need passing information
to the method

1. Parameters and arguments

Parameters:
Variables in the parentheses of our method
that can then use inside our method

```
public void greeting(){  
    System.out.println("Hello Daniel");  
}
```

String name

1. Parameters and arguments

Parameters:
Variables in the parentheses of our method
that can then use inside our method

```
public void greeting(String name){  
    System.out.println("Hello "+name);  
}
```

1. Parameters and arguments

~~greeting();~~

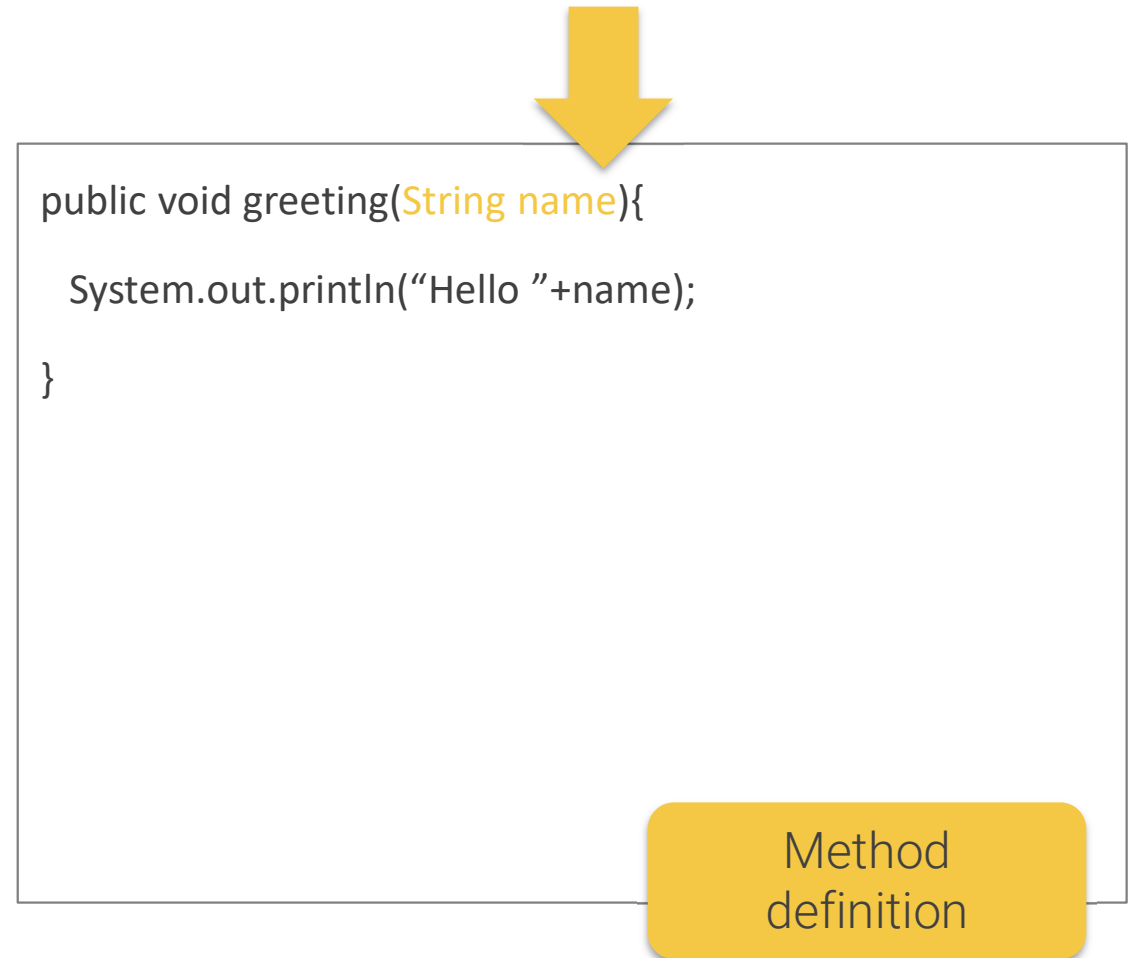
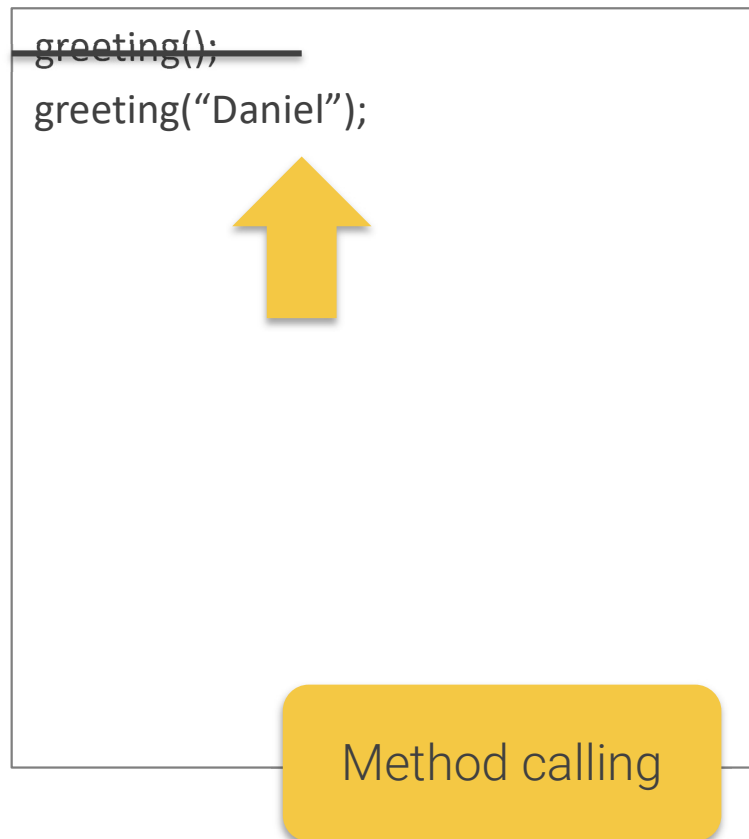
Compilation error

Method calling

```
public void greeting(String name){  
    System.out.println("Hello "+name);  
}
```

Method
definition

1. Parameters and arguments



1. Parameters and arguments

Arguments:
Specific values passed into our method call

```
greeting("Daniel");
```

Method calling

```
public void greeting(String name){  
    System.out.println("Hello "+name);  
}
```

Method
definition

1. Parameters and arguments

Parameters:
Variables in the parentheses of our method
that can then use inside our method

```
greeting("Daniel");
```

Method calling

```
public void greeting(String name){  
    System.out.println("Hello "+name);  
}
```

Method
definition

1. Parameters and arguments

Arguments:
Specific values passed into our method call

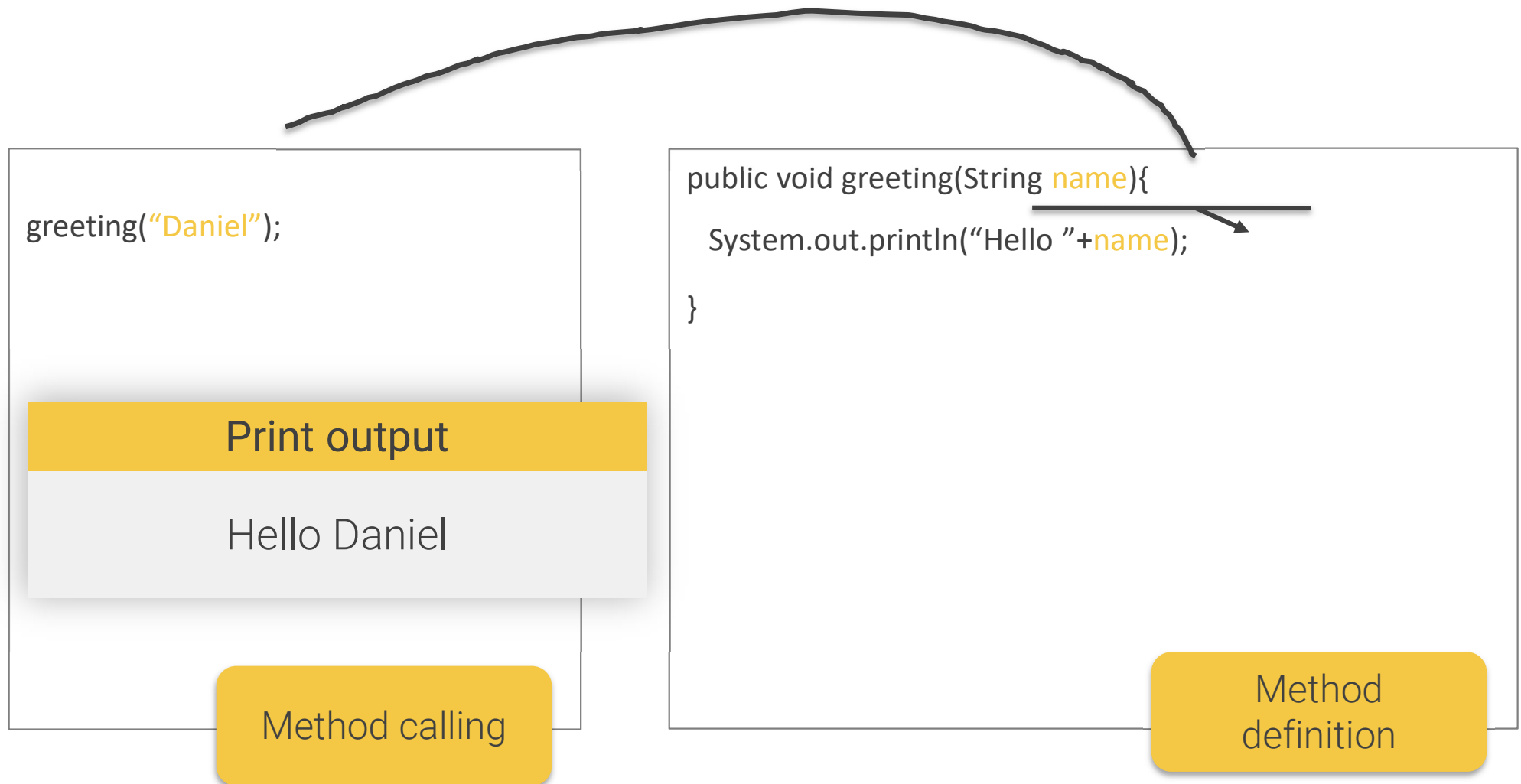
```
greeting("Daniel");
```

Method calling

```
public void greeting(String name){  
    System.out.println("Hello "+name);  
}
```

Method
definition

1. Parameters and arguments



1. Parameters and arguments

```
greeting();  
greeting("Daniel");  
greeting(3);
```

Compilation error

Method calling

```
public void greeting(String name){  
    System.out.println("Hello "+name);  
}
```

Method
definition

1. Parameters and arguments

```
greeting();  
greeting("Daniel");  
greeting(3);  
greeting("Sara");
```

Print output

Hello Daniel
Hello Sara

Method calling

```
public void greeting(String name){  
    System.out.println("Hello "+name);  
}
```

Method
definition

¿Cual es la salida por consola con la llamada *interpreta(5)* ?

```
public void interpreta(int x){  
    if (x>25){  
        System.out.println("Hace calor");  
    }else if (x>10){  
        System.out.println("No hace ni frío ni calor");  
    }else{  
        System.out.println("Hace frío");  
    }  
}
```

OUTPUT:

1. Parameters and arguments

¿Cual es la salida por consola con la llamada *interpreta(5)* ?

```
public void interpreta(int x){  
    if (x>25){  
        System.out.println("Hace calor");  
    }else if (x>10){  
        System.out.println("No hace ni frío ni calor");  
    }else{  
        System.out.println("Hace frío");  
    }  
}
```

x=5

OUTPUT:

1. Parameters and arguments

¿Cual es la salida por consola con la llamada *interpreta(5)* ?

```
public void interpreta(int x){  
    if (x>25){  
        System.out.println("Hace calor");  
    }else if (x>10){  
        System.out.println("No hace ni frío ni calor");  
    }else{  
        System.out.println("Hace frío");  
    }  
}
```

x=5

OUTPUT:

1. Parameters and arguments

¿Cual es la salida por consola con la llamada *interpreta(5)* ?

```
public void interpreta(int x){  
    if (x>25){  
        System.out.println("Hace calor");  
    }else if (x>10){  
        System.out.println("No hace ni frío ni calor");  
    }else{  
        System.out.println("Hace frío");  
    }  
}
```

x=5

OUTPUT:

1. Parameters and arguments

¿Cual es la salida por consola con la llamada *interpreta(5)* ?

```
public void interpreta(int x){  
    if (x>25){  
        System.out.println("Hace calor");  
    }else if (x>10){  
        System.out.println("No hace ni frío ni calor");  
    }else{  
        System.out.println("Hace frío");  
    }  
}
```

x=5



OUTPUT:

Hace frío

¿Y con la llamada *interpreta(20)* ?

```
public void interpreta(int x){  
    if (x>25){  
        System.out.println("Hace calor");  
    }else if (x>10){  
        System.out.println("No hace ni frío ni calor");  
    }else{  
        System.out.println("Hace frío");  
    }  
}
```


OUTPUT:

1. Parameters and arguments

¿Y con la llamada *interpreta(20)* ?

```
public void interpreta(int x){  
    if (x>25){  
        System.out.println("Hace calor");  
    }else if (x>10){  
        System.out.println("No hace ni frío ni calor");  
    }else{  
        System.out.println("Hace frío");  
    }  
}
```

x=20




OUTPUT:

1. Parameters and arguments

¿Y con la llamada *interpreta(20)* ?

```
public void interpreta(int x){  
    if (x>25){  
        System.out.println("Hace calor");  
    }else if (x>10){  
        System.out.println("No hace ni frío ni calor");  
    }else{  
        System.out.println("Hace frío");  
    }  
}
```

x=20




OUTPUT:

1. Parameters and arguments

¿Y con la llamada *interpreta(20)* ?

```
public void interpreta(int x){  
    if (x>25){  
        System.out.println("Hace calor");  
    }else if (x>10){  
        System.out.println("No hace ni frío ni calor");  
    }else{  
        System.out.println("Hace frío");  
    }  
}
```

x=20




OUTPUT:

1. Parameters and arguments

¿Y con la llamada *interpreta(20)* ?

```
public void interpreta(int x){  
    if (x>25){  
        System.out.println("Hace calor");  
    }else if (x>10){  
        System.out.println("No hace ni frío ni calor");  
    }else{  
        System.out.println("Hace frío");  
    }  
}
```

x=20



OUTPUT:

Hace hace ni frío ni calor

Variable parameters

2. Variable parameters

```
greeting("Daniel");
```

Print output

Hello Daniel

Method calling

```
void greeting(String name){  
    System.out.println("Hello "+name);  
}
```

Method
definition

2. Variable parameters

```
String passengerName= "Daniel";  
greeting(passengerName);
```

Print output

Hello Daniel

Method calling

```
public void greeting(String name){  
    System.out.println("Hello "+name);  
}
```

Method
definition

2. Variable parameters

```
String passengerName= "...";  
...  
passengerName = "Sara";  
greeting(passengerName);
```

Print output

Hello Sara

Method calling

```
void greeting(String name){  
    System.out.println("Hello "+name);  
}
```

Method
definition

“No juzgues cada día por la cosecha que recoges,
sino por las semillas que plantas .”

Robert Louis Stevenson, novelista poeta y ensayista escocés

