

Class Methods

Methods that don't go with objects –
class methods

Methods with Robots

- Methods are defined in a class. They give objects extra power, new instructions, e.g. `turnRight()`.
- You need to create an instance of an object before you can access the methods

```
Robot bob;  
bob.turnRight();
```

Error!

```
Robot bob;  
bob = new Robot();  
bob.turnRight();
```

Methods with Robots (2)

- You can pass information to a method:

```
jim.run(4);
```

```
world.addBeeper(2, 9);
```

- You can receive information from a method

```
x = bob.countToWall();
```

So what more do you need?

- The vast majority of methods that you write will be associated with an object.
- But ...many programs also have methods that don't naturally belong to an object: these are called class methods (as opposed to instance methods)
- Defined using the keyword **static**.

Example: square

Here is a method that returns the square of a number

```
int square(int num)  
{  
    int s = num * num;  
    return s;  
}
```

This method calculates the square of a number and returns it.

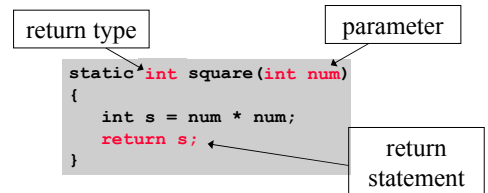
Problem

1. To write the method, we need an appropriate class
2. We need to create an object before we can invoke the method.

Solution

- Use a class method: A class method is not associated with any object.
- Class methods are introduced with the static keyword.

Static Methods



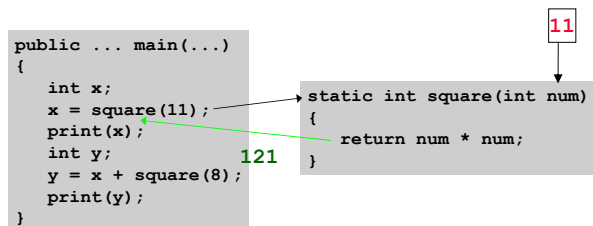
The square method

The return statement can take any expression of the return type (in this case int). So the following method definition is equivalent:

```
static int square(int num)
{
    return num * num;
}
```

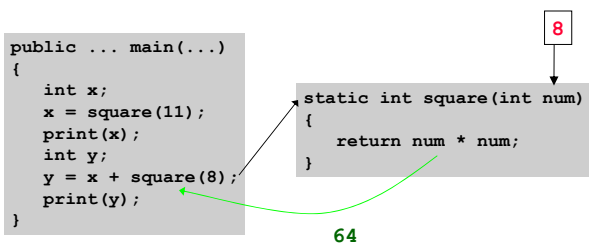
Invoking the square method

The square method can be used as follows



Invoking the square method

The square method can be used as follows



Summary

- Some methods naturally do not belong to any object.
- Make them class methods using the static keyword.
- You don't need to create an object in order to be able to use them.