Static vs. Instance Methods

Static vs. Instance Methods

Static Methods

Static methods are declared using a static modifier.

Static methods can't access instance methods and instant variables directly.

They're usually used for operations that don't require any data from an instance of the class (from '**this**').

If you remember, the this keyword is the current instance of a class.

Static Methods

So inside a static method, we can't use the **this** keyword.

Whenever you see a method that doesn't use instance variables, that method should probably be declared as a static method.

For example, main is a static method, and it's called by the Java virtual machine when it starts the Java application.

Static Methods Example

```
class Calculator {

    public static void printSum(int a, int b) {
        System.out.println("sum= " + (a + b));
    }
}

public class Main {

    public static void main(String[] args) {
        Calculator.printSum(5, 10);
        printHello();   // shorter from of Main.printHello();
    }

    public static void printHello() {
        System.out.println("Hello");
    }
}
```

static methods are called as **ClassName.methodName();** or **methodName();** only if in the same class

In this example
Calculator.printSum(5,10);
printHello();

Instance Methods

Instance methods belong to an instance, of a class.

To use an instance method, we have to instantiate the class first, usually by using the **new** keyword.

Instance Methods

Instance methods can access instance methods and instance variables directly.

Instance methods can also access static methods and static variables directly.

Instance Method Example

```
class Dog {

    public void bark() {
        System.out.println("woof");
    }
}

public class Main {

    public static void main(String[] args) {

        Dog rex = new Dog();                    // create instance
        rex.bark();                             // call instance method
    }
}
```

Static or Instance Method