Xueyang Huang

Exercise 1:
(The first line is the method header)
public static (the modifier) String (Return value type) method(int n1, String s1 (these two are formal parameters)) (Method name + Parameter list)
{
(The following is the method body)
	return s1 + " " + n1; (return value)
}

Exercise 2:

```
public class Test {
	public static void main(String[] args) {
		System.out.println(max(1, 2.0));
  }
	public static double max(int num1, double num2) {
		if (num1 > num2)
			return num1;
		else
			return num2;
}
	public static double max(double num1, int num2) {
		if (num1 > num2)
			return num1;
		else
			return num2;
	}
}
```

It does compile. Because when the max method is invoked in the main method, the datatypes of the two values inside the parameter list are an integer and a floating point value. Even if there are two max methods inside the program, the datatypes of the parameters in each of the methods' parameter lists are different. The datatypes of the parameters in the main method only correspond to the first max method created in the program, therefore there will not be an ambiguous invocation. To break it, just turn the floating point number "2.0" in the main method to integer "2".