

# Exercises on Methods

COMP-202B, Winter 2011, All Sections

## Methods and Method Headers

Fill in the following chart with the information about the name of the method, types of the input variables, and the return type. The first one is done for you

Method header	Method name	Types of Input Variables	return type
public static void fun(int x, double y)	fun	one int, one double	void
public static int silly(int x)			
public static int silly(int x)			
public static float sunny(boolean b, Flamingo f)			
public void day()			
private static Chair house(int x, double d, char s, String a)			

## Passing input to methods

Suppose you are given the following methods:

```
public static void fun(int x, double y) {  
    System.out.println("I don't return anything :( ");  
}
```

```
public static double hello(double y) {  
    return y * 2;  
}
```

```
public static double mystery(int x, int y) {  
    return x * y;  
}
```

Each row of the table has a small piece of code. You should circle whether it compiles or not. If it compiles, you should write the name and values of all variables inside the *outer most* method that are already initialized. If it does not compile write down the reason why.

Then write down the value that the method returns. Remember that this value is the value that would be used if the call to the method was part of another computation (for example inside `System.out.println()` or storing something in a variable, or part of an arithmetic expression)

Command to call	Compiles?	Initialized Variables in Method	return value
<code>fun(3,4.0);</code>	<b>yes</b> / no	<code>x = 3, y = 4</code>	void
<code>hello(3);</code>	<b>yes</b> / no	<code>y = 3</code>	6
<code>int x= fun(3-2,4.0);</code>	yes/ <b>no</b>	fun returns void. Can't store result into int variable	N/A
<code>fun(34.0);</code>	yes/ <b>no</b>	too few arguments	N/A
<code>fun(3, hello(4));</code>	<b>yes</b> / no	<code>x = 3, y = 8</code>	void
<code>fun(3-2,4.0);</code>	yes/ no		
<code>double x = 3; fun(3 + 1, x);</code>	yes/ no		
<code>fun(3,4.0);</code>	yes/ no		
<code>double hello = Math.pow(2,2); fun(foo);</code>	yes/ no		
<code>hello(Math.pow(2,2));</code>	yes/ no		
<code>hello(Math.pow( Math.pow(2,2), Math.pow(2,2)));</code>	yes/ no		
<code>int x = hello(3);</code>	yes/ no		
<code>double x = hello(3);</code>	yes/ no		
<code>mystery(4, foo(3,4));</code>	yes/ no		
<code>mystery(4, hello(3,4));</code>	yes/ no		
<code>mystery(4, mystery(3,4));</code>	yes/ no		
<code>double x = 1; mystery(4, mystery(x,x));</code>	yes/ no		
<code>double x = 1; int y = x; mystery(4, mystery(y,y));</code>	yes/ no		
<code>double x = 1; mystery(4, hello(x));</code>	yes/ no		