

main()

Remember that a program in Java is a collection of classes. You run a program by telling Java, or Eclipse to run one of the program's classes.

The system expects the class you are running to contain a method called `main()`.

The `main()` method must be declared both *public* and *static*. It should be public because you are running it outside its class. When a variable or method is declared static, that means there is only one copy for the class and it can be accessed without creating an object of the class. If you had a class *Person* and wanted to have a variable *population* for the class that was incremented every time you construct a new object of the class, you would want that variable to be declared static. You could refer to that variable as `Person.population`; you don't need an object of the class for this.

Similarly, if you tell java to run class `Person`, the system will call `Person.main()`

The full signature of the `main()` method is
`public static void main(String [] args)`

The parameter *args* means that `main` can be given an array of `String` arguments. In Eclipse we specify the arguments in the *Arguments* tab of the Run Configuration window.

For example, if you want program `PrintFile` to work with file `foobar.txt` you would put `String "foobar.txt"` as its argument in the Run Configuration and inside the program refer to the file to be printed as `args[0]`

The arguments to a program are `args[0]`, `args[1]`, `args[2]`, etc.

`args.length` is the number of arguments the program was actually called with, whether they are used in the program or not.

The fact that `main()` must be a static method sometimes gives Java beginners fits. Static means it can be used outside the class without an object of the class. `main()` can only directly call methods of the class that are themselves declared static.

So if we have a class `Primes` that has a `main()` method that prints out a list of primes, `main()` could call a method

```
static boolean isPrime(int x)
```

to determine if number `x` is prime. This method can be declared static because you don't need any properties of the `Primes` class to decide if `x` is prime. `main()` can't call `isPrime()` if it isn't declared static.

On the other hand, if you have a class Student that has properties such as name, age, id, gpa, classList, etc for a student, a method such as setAge() could not be declared static. This method would be setting the age property of a specific student object.

main() in this case, would work by creating Student objects. It sometimes surprises beginners to Java that methods of a class such as Student can themselves construct Student objects.