Java basics

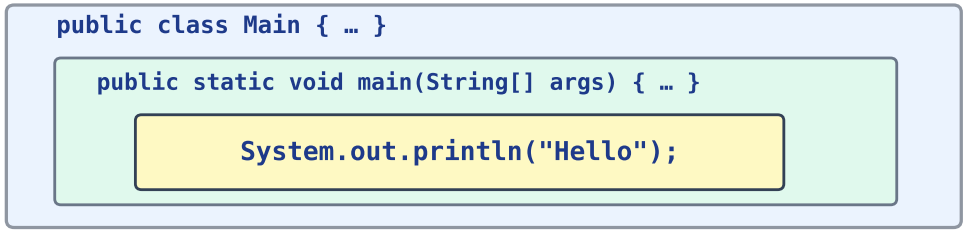
Java Reference

Classes, main & output

This site compiles Java using Java 8 language rules. These standalone examples put their entry method , `main`, inside a class . Helper classes and method-only answers do not need their own `main`. `System.out.println(...)` prints a line; `System.out.print(...)` prints with no new line.

```
public class Main {
    public static void main(String[] args) {
        System.out.println("Hello, world!");
        System.out.println("I am learning Java.");
    }
}
```

- Java is compiled : the compiler checks the whole program, then it runs.
- Declarations, assignments, method calls and `return` statements end with a semicolon `;`. An `if`, `for` or `while` block uses braces; do not add a semicolon just because its closing brace appears.



Exam tip: every program lives in a class; execution starts at main; end each statement with ;

This standalone Java 8 example defines main inside a class

Variables & primitive types

A variable must declare its type. Common primitive types : `int` (whole number), `double` (decimal), `boolean` (`true`/`false`), `char` (one UTF-16 code unit; some characters need two).

```
public class Main {
    public static void main(String[] args) {
        int age = 17;
        double price = 9.99;
        boolean passed = true;
        System.out.println(age + " " + price + " " + passed);
    }
}
```

Comments & style

A comment is `//` (one line) or `/* ... */` (a block). Indent the code inside braces `{ }`. Class names start Capitalised; variables and methods use camelCase (lowercase first).

```
public class Main {
    public static void main(String[] args) {
        // greet the user
        String firstName = "Mei";
        System.out.println("Hi, " + firstName);
    }
}
```

Common mistakes

- Declarations, assignments, method calls and `return` statements end with `;`; control-flow blocks do not need an extra semicolon.
- Use the standard Java 8 entry point `public static void main(String[] args)`; the parameter name may differ, and `String... args` is also valid.
- `println` adds a new line; `print` does not.