

5 main() Method

1. What is the signature of main method?

- `public static void main(String[] args)`
- Signature = `main(String[] args)`
- (Return type and access modifier are not part of the signature.)

2. Why main method is public?

- The JVM must access the `main()` method from outside the class.
- If it is not public, JVM cannot call it and the program won't start.

3. Why main method is static?

- The JVM calls `main()` without creating an object of the class.
- Static methods belong to the class, not to objects — so no object creation is needed.

4. Why main return type is void?

- The JVM does not expect any value back after program execution.
- So `main()` returns nothing, hence void.

5. Can we change main method signature?

- ❌ No
- If you change:
 - method name
 - parameters
- JVM will not recognize it as the entry point and gives: *"Main method not found"* error.
- Only valid variations:
 - `public static void main(String[] args)`
 - `public static void main(String args[])`
 - `public static void main(String... args)`

6. Can we overload main method?

- Yes, we can overload it.
 - ✓ `public static void main(int a) { }`
- But JVM will only call:
 - ✓ `main(String[] args)`
- Other overloaded methods must be called manually.

7. Can we make main method final?

- Yes
- Making main() final prevents it from being overridden,
- but JVM execution is not affected.

8. Can we make main method private?

- No
- If main() is private, JVM cannot access it → program will not run.

9. Can Java program run without main?

- Yes, but only in special cases:
- Using static blocks (older Java versions)
- Using frameworks like:
 - Spring Boot
 - JavaFX
 - Servlet containers
- For a normal Java application, main() is mandatory.

10. What happens if main is not static?

- Program will not run
- Reason: JVM cannot create an object automatically to call a non-static method.