Féidearthachtaí as Cuimse
Infinite Possibilities

Static in Java

Object Oriented programming

OLLSCOIL TEICNEOLAÍOCHTA
BHAILE ÁTHA CLIATH
TU DUBLIN
TECHNOLOGICAL
UNIVERSITY DUBLIN

“static”   keyword : for attributes

- In Java, the static keyword is used to **define class-level members** (variables, methods, or blocks) that **belong to the class itself** rather than to any specific instance of the class

- Static attributes .. Data stored in static variables is common for all the objects (or instances) of that Class.

- Examples: pi, currentStudentNumber etc.
  - Refer to them using the class ref
    - E.g. Circle.pi

Going deeper

When you make a variable **static**, it belongs to the **class itself**, not to any individual **object** made from that class.

So:

- If you create 10 objects of the same class, they will **all share** the same static variable.
- If one object changes that variable’s value, **all the others see the change** too.

```
class Student {

   int studentNumber;

   static int studentCount = 100; // Shared across all instances

       public Student() {


       // .. What code do I need here to make the allocated student
             numbers be 100,101,102,203…e tc

        }



Increments every time a new Student is created }
```

```
class Student {

   int studentNumber;

   static int studentCount = 100; // Shared across all instances

       public Student() {

             studentNumber = studentCount;
             studentCount++;                // increment

}
```

“static”   keyword : for methods

- Remember - In Java, the static keyword is used to **define class-level members** (variables, methods, or blocks) that **belong to the class itself** rather than to any specific instance of the class

- Static methods can be used when a method does not use any object (instance information)
  - “Does it make sense to call this method, even if no object has been constructed yet or exists?" If so, make it static”

Example

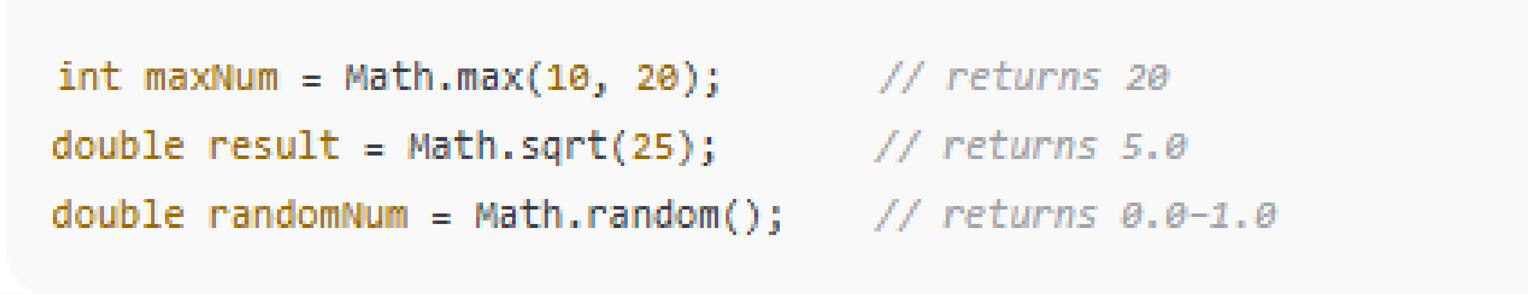
```
public static double convertKmToMiles(double km) {
    return km * 0.621371;
}


double miles = Sports.convertKmToMiles(10);
```

“static” keyword : for methods

- E.g `main` method
  - We haven’t been instantiating the control class it is in…
- Utility methods
  - E.g. a Sports class with method to convert kilometres to miles

Math

- Utility/helper methods



```
int maxNum = Math.max(10, 20);        // returns 20
double result = Math.sqrt(25);        // returns 5.0
double randomNum = Math.random();     // returns 0.0-1.0
```

Correction

- String was the original immutable type — and now LocalDate follows the same principle.

```
// Mutable
Date d = new Date();
d.setYear(2005);   // changes the same object


// Immutable
LocalDate ld = LocalDate.of(2005, 1, 1);
ld.withYear(2010); // returns NEW LocalDate
```

Factory Methods

- A **factory method** is a static method that creates and returns an object —
  instead of using a public constructor (new) directly.

LocalDate Uses Static Factory Methods

- LocalDate today = LocalDate.now();   // current system date
- LocalDate birthday = LocalDate.of(1999, 12, 25);   // specific date
- LocalDate parsed = LocalDate.parse("2025-10-18");

- Java **hides the constructors** and instead provides static factory methods to build new objects safely.

Exam Paper 2024

Explain the effect of the Java keyword static when applied to each of the following two Java statements:

(i) private static multiplierFactor = 100;

(ii) public static void calcTax(int month);

Answers

I) Shared by all objects – there is only **one copy** of this variable in the whole class, not one per object.

•It stays in memory **as long as the class exists**.

•You can use it without creating an object, for example:

II) Class method: callable without an instance: ClassName.calcTax(…).

•No this: cannot refer to instance fields/methods directly (only static members or parameters).

•Typical use: utility, pure/functional calculations, factory helpers, etc.