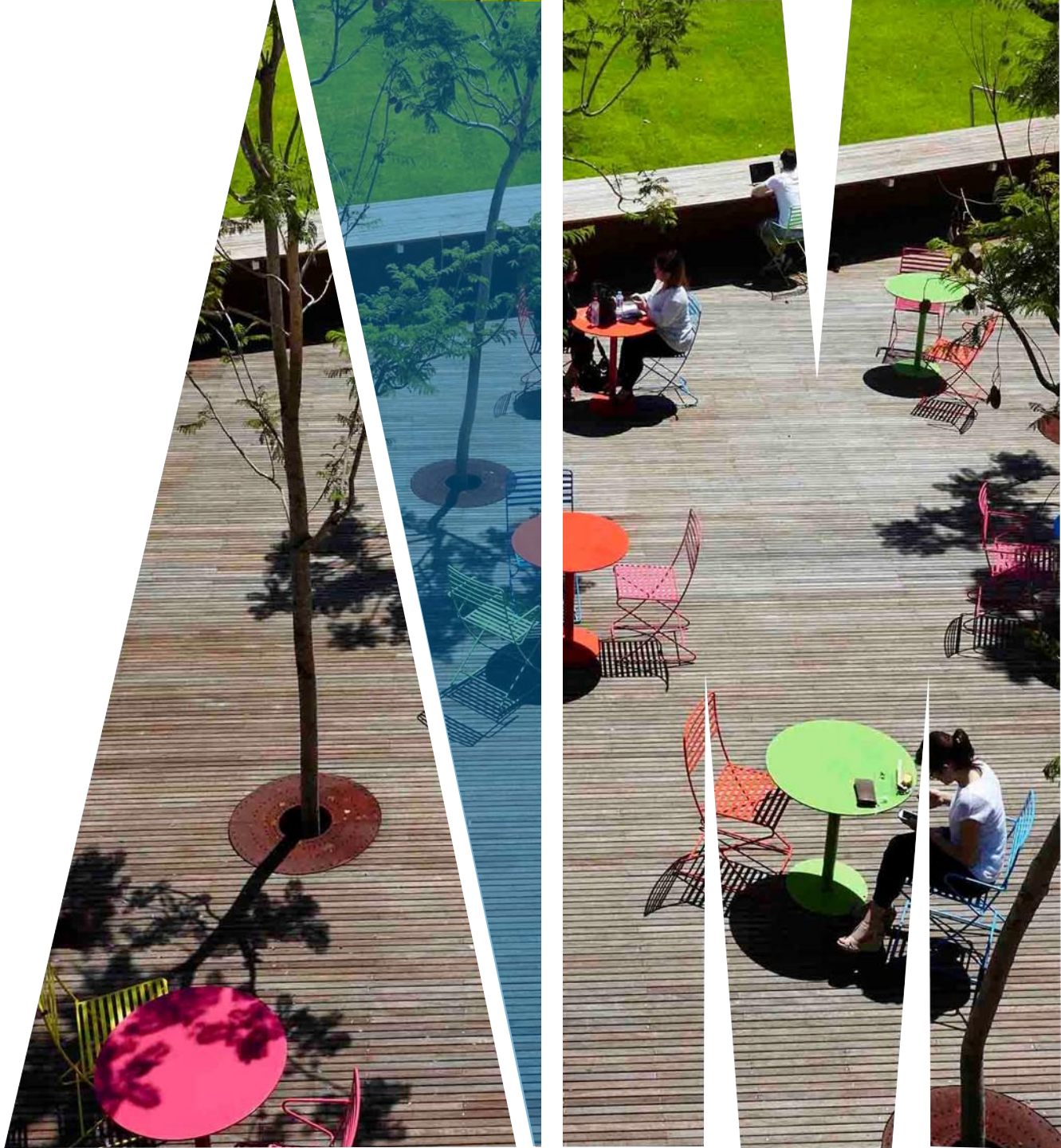
MONASH University

FIT2099 Object-Oriented Design and Implementation

Static and final in Java



MONASH University

Outline

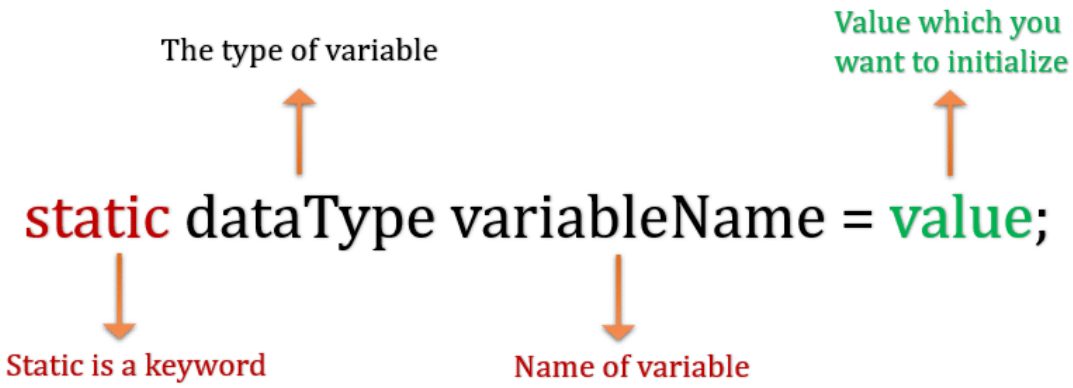
Static keyword

Final keyword

WHAT IS THE STATIC KEYWORD?

The **static keyword** in Java is used to refer to the common property of all objects (which is not unique for each object).

The users can apply static keywords with variables, methods, blocks, and nested classes.

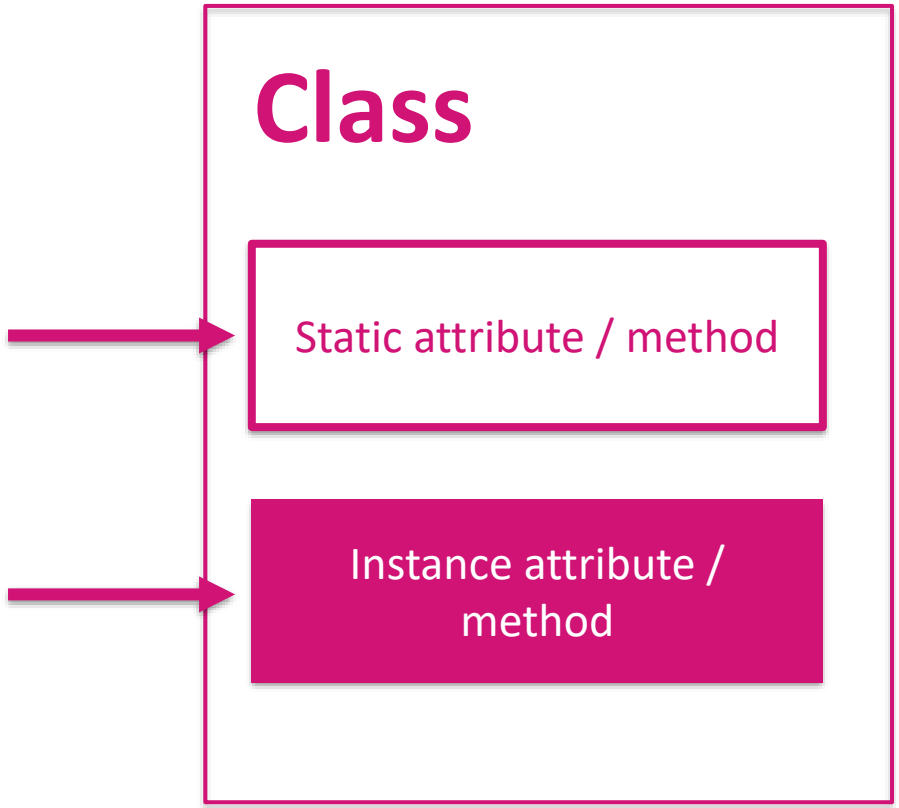


WHAT IS A STATIC CLAS MEMBER?

When a class member is declared with static, it is **not required to create an object** to call methods and attributes.

They can be accessed without creating an instance (object)

They **cannot** be accessed **without** creating an instance (object)



HOW TO ACCESS A STATIC CLAS MEMBER?

Instead, **the class name** can be used to call it.

```
1 class Student{
2     int studentid; //instance variable
3     String name; //instance variable
4     static String university ="Monash University"; //static variable
5
6     Student(int _studentid, String _name){
7         studentid = _studentid;
8         name = _name;
9     }
10
11    void display (){System.out.println(this.studentid+" "+this.name+" "+university);}
12 }
```

HOW TO ACCESS A STATIC CLAS MEMBER?

Instead, **the class name** can be used to call it.

```
1 class Student{
2     int studentid; //instance variable
3     String name; //instance variable
4     static String university ="Monash University"; //static variable
5
6     Student(int _studentid, String _name){
7         studentid = _studentid;
8         name = _name;
9     }
10
11    void display (){System.out.println(this.studentid+" "+this.name+" "+university);}
12 }
```

HOW TO ACCESS A STATIC CLAS MEMBER?

Instead, **the class name** can be used to call it.

```java
class Student{
    int studentid; //instance variable
    String name; //instance variable
    static String university ="Monash University"; //static variable

    Student(int _studentid, String _name){
        studentid = _studentid;
        name = _name;
    }

    void display (){System.out.println(this.studentid+" "+this.name+" "+university);}
}
```

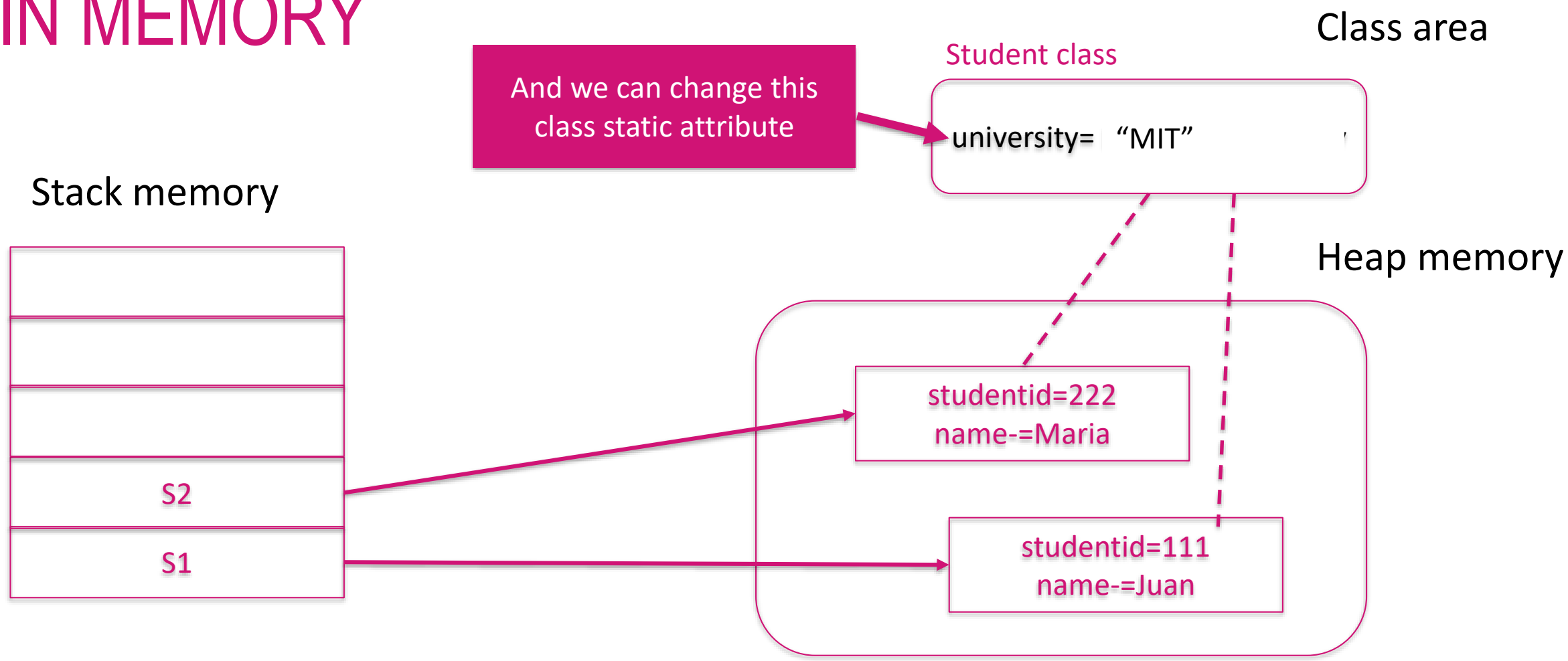
```java
public class Test{
    public static void main(String args[]){
        Student s1 = new Student(111,"Juan");
        Student s2 = new Student(222,"Maria");
        //we can change the college of all objects by the single line of code
        //Student.university="MIT";
        s1.display();
        s2.display();
    }
}
```

Output:

```
111 Juan Monash University
222 Maria Monash University
```

MONASH University

A STATIC CLASS MEMBER
IN MEMORY



HOW TO ACCESS A STATIC CLAS MEMBER?

Output:

```
111 Juan MIT
222 Maria MIT
```

```java
public class Test{
    public static void main(String args[]){
        Student s1 = new Student(111,"Juan");
        Student s2 = new Student(222,"Maria");
        //we can change the college of all objects by the single line of code
        Student.university="MIT";
        s1.display();
        s2.display();
    }
}
```

WHY DO WE USE STATIC CLAS MEMBERS AGAIN?

Some advantages (with caveats):

Can do **meta object operations** (like validating something before creating objects, keep count of number of objects).

Can do **operations which have nothing to do with objects** but still you want them to be tied to Class.

Effective heap **memory use**.

WHY SHOULD WE USE STATIC CAREFULLY?

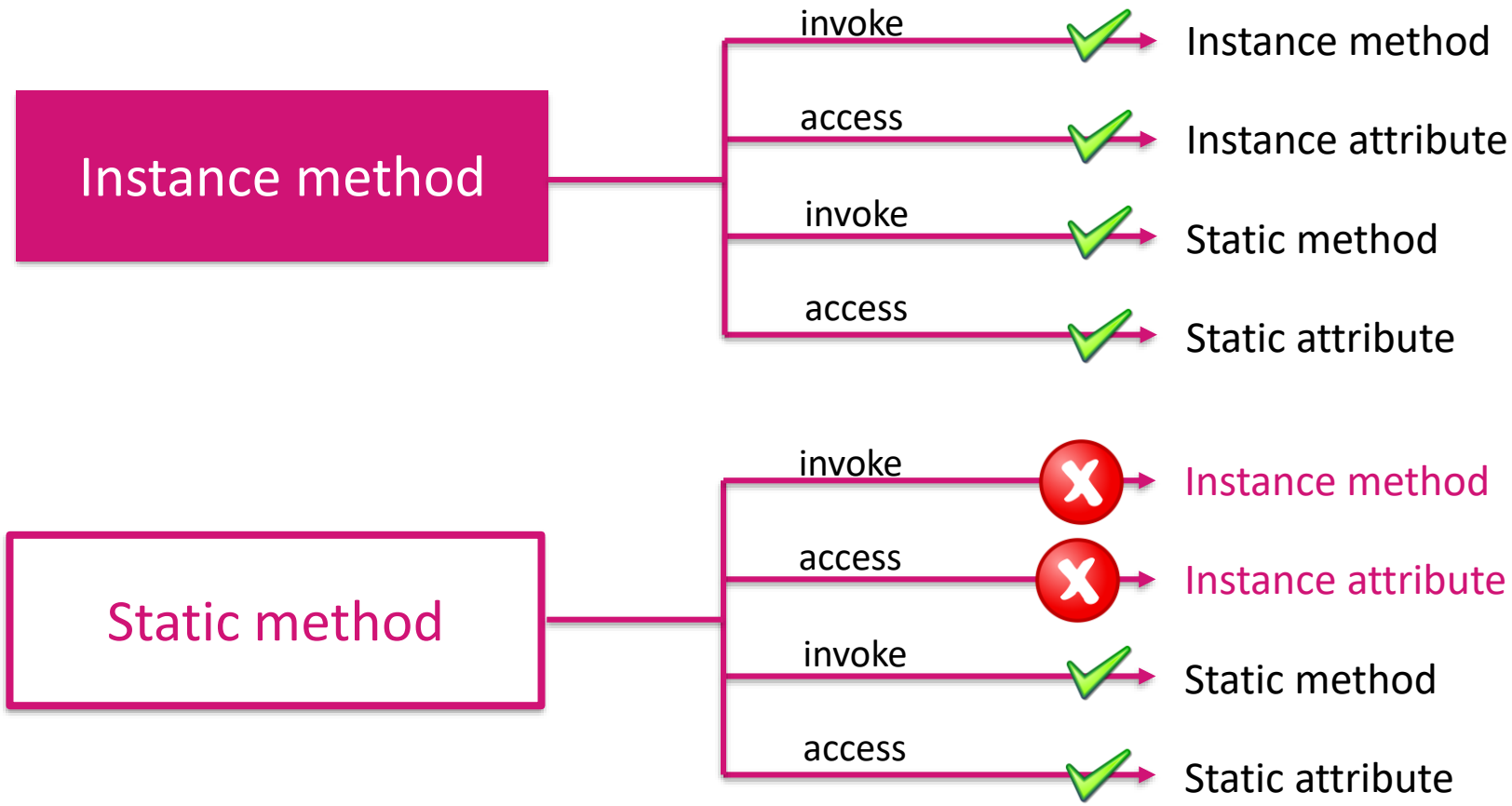
Some dis-advantages:

Doesn't reflect pure **Object Oriented approach** (disregards encapsulation: an object is no longer in complete control of its state).

Can lead to **conflicts** if updated by multiple objects and threads.

Remain in memory until the application terminates.

CHEATSHEET FOR INSTANCE AND STATIC MEMBERS IN THE SAME CLASS



WHAT IS THE FINAL FOR VARIABLES?

Variables marked as final can't be reassigned. Once a final variable is initialized, it can't be altered. It is a way to declare "constant" values.

```
1 public void finalVariableAssignedOnlyOnce() {
2     final int i = 1;
3     //...
4     i=2;
5 }
```

Compiler error!

WHAT IS THE FINAL FOR VARIABLES?

It is a way to declare “constant” values.

```
static final int MAX_WIDTH = 999;
```

THE FINAL KEYWORD for classes

If you don't want other classes to inherit from a class, use the **final** keyword:

```
final class Vehicle {
  ...
}

class Car extends Vehicle {
  ...
}
```

If you try to inherit from a final class, Java will throw an error

…and if used with methods these cannot be overridden.

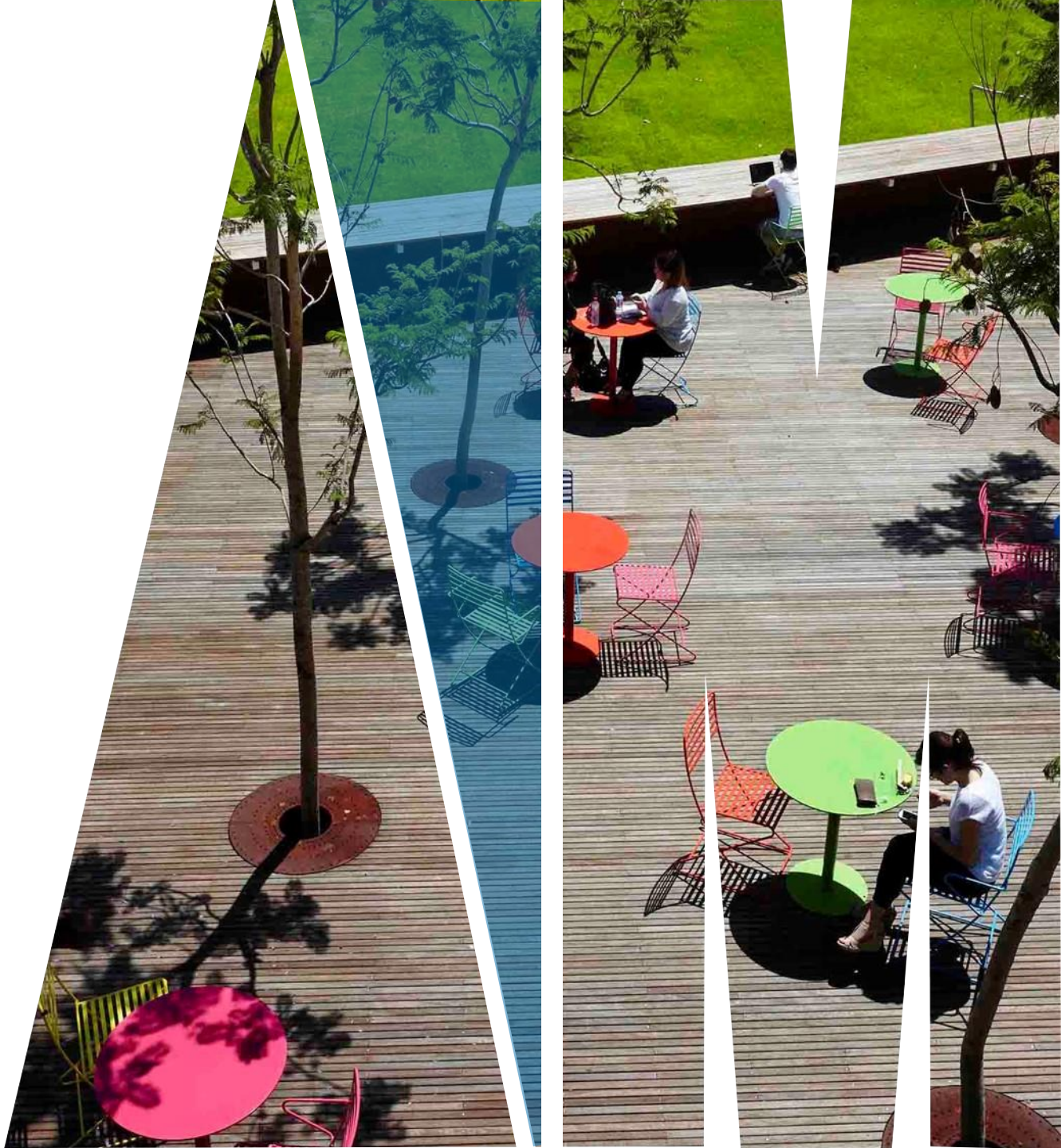
Summary

Static keyword

Final keyword

MONASH University

Thanks



MONASH University