Mohammed Roshan Khan

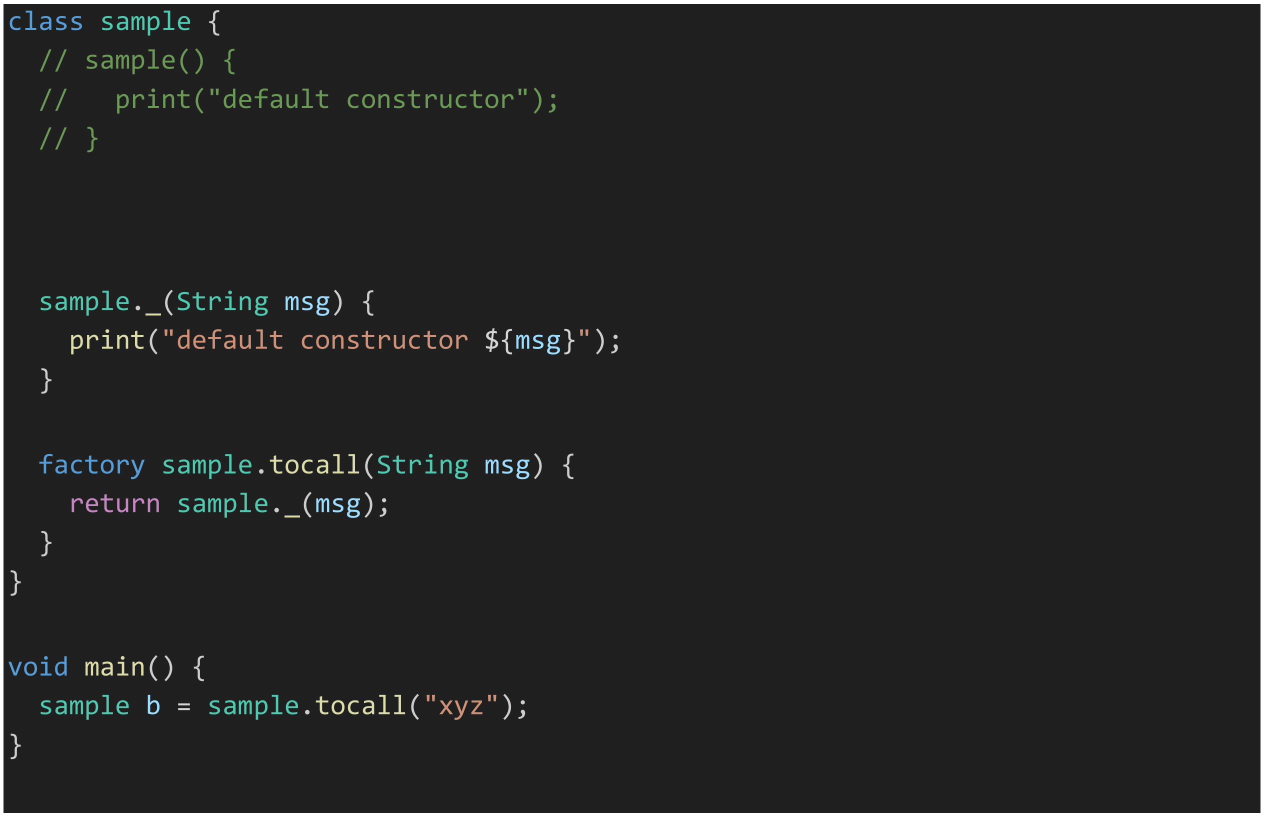
Day6 – Assignment

Private constructor

In Dart, a **private constructor** is a constructor that can only be accessed within the same library. It is declared by prefixing the constructor's name with an underscore (_). This is often used to:

- Restrict object creation from outside the class.
- Implement singleton patterns.
- Control instance creation.

Ex:-



```
class sample {
  // sample() {
  //   print("default constructor");
  // }



  sample._(String msg) {
    print("default constructor ${msg}");
  }

  factory sample.tocall(String msg) {
    return sample._(msg);
  }
}

void main() {
  sample b = sample.tocall("xyz");
}
```

Getters & Setters

- You make a **getter** or **setter** private by prefixing its name with an underscore _.
- Private getters/setters can only be accessed **within the same library**.
- They're useful to control access to class properties and encapsulate internal data.

Ex:-

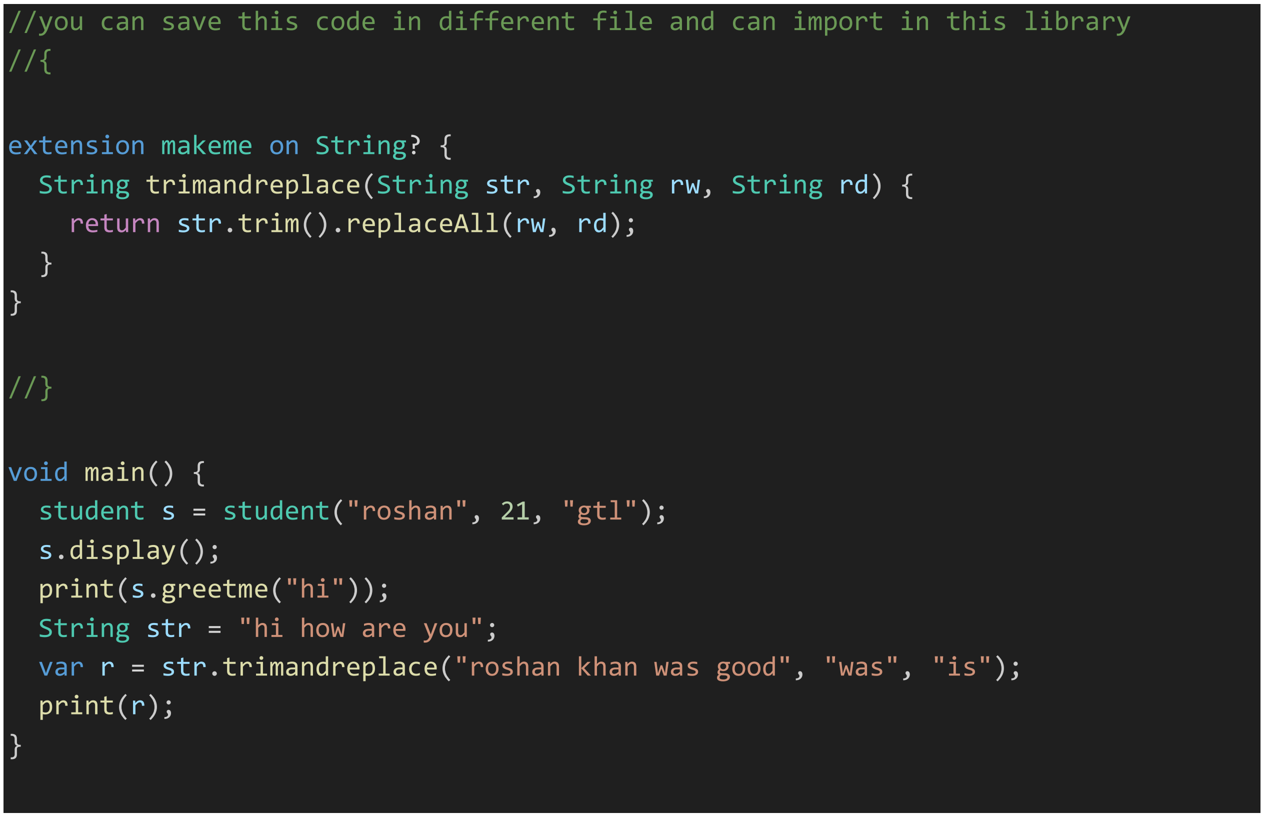
```dart
class person {
  String name = "";
  int _age = 0;

  void setName(String name) {
    this.name = name;
  }

  void setAge(int age) {
    this._age = age;
  }

  String get getName => name;
  // or you can use this as well
  // String getName() {
  //   return name;
  // }

  int getAge() {
    return _age;
  }
}

void main() {
  person p = person();
  p.setName("roshan");
  print(p.getName);
  p.setAge(22);
  print(p.getAge());
}
```

Extension in Dart

- Extensions themselves **cannot be made private**, but **their members (methods/getters/setters) can** be private by prefixing with _.
- However, if you **name the extension with a leading underscore**, it becomes private to the library.
- Private extension members and private extension names are only accessible **within the same library**.

Ex:-

```
/*syntax of the extention methods

extention Extensionname on classname{
    returntype methodname(parameters)
    {
      statements;
    }
}*/

class student {
  String? name;
  int? age;
  String? city;

  student(String name, int age, String city) {
    this.name = name;
    this.age = age;
    this.city = city;
  }

  void display() {
    print("${this.name} ${this.age} ${this.city}");
  }
}

extension greetings on student {
  String greetme(String str) {
    return ("$str ${this.name}");
  }
}
```



```dart
//you can save this code in different file and can import in this library
//{

extension makeme on String? {
  String trimandreplace(String str, String rw, String rd) {
    return str.trim().replaceAll(rw, rd);
  }
}

//}

void main() {
  student s = student("roshan", 21, "gtl");
  s.display();
  print(s.greetme("hi"));
  String str = "hi how are you";
  var r = str.trimandreplace("roshan khan was good", "was", "is");
  print(r);
}
```