Late variable:

Late variable is used when you don't want to initialize a variable immediately but sometimes you want to delay the initialization.

```
class sample{
late String name;   //Late variable

void setup() {
  name = "Samudrala Tarunika";
  print(name);
}
}
void main() {
  sample s1=sample();
  s1.setup();
}
```

Null Assertion:

If variable is actually null at runtime it throws an exception.

```
String? name= null;
print(name!.length);    / / Runtime error!  Can't call length on null
```

Null Aware:

No runtime error if variable is null.

```
String? name = null;
print(name?.length); // Output: null (no error)
```

this:

1. This keyword to disambiguate between instance variables and parameters with the same name:

```
class Person {
  String name;

  Person(String name) {
    this.name = name;
  }
}
```

2. Named constructor using this keyword

```
class Person {
  String name;
  int age;

  Person(this.name, this.age);

  Person.withName(this.name) : age = 0;
}
```

3. Here, this returns the current instance

```
class Counter {
  int count = 0;

  void increment() {
    count++;
  }

  Counter getCounter() {
    return this;    }
}
```

Private variables:

Private variables to a library or class are not accessible outside of that library.

Syntax:
data_type variable_name;
String _accountNumber;

Factory Constructor:

```
class MyClass {
  String name;

  MyClass._private(this.name);  // Private constructor
  factory MyClass.create(String name) {
    if (name.isEmpty) {
      throw Exception("Name can't be empty");
    }
    return MyClass._private(name);  // Using private constructor here
  }
}
void main(){
  MyClass m1=MyClass.create("Tarunika");
  print(m1.name);
}
```

Factory named constructor:

```
class MyClass {
  String? name;
  int? age;

  MyClass._private({ this.name,  this.age});

  factory MyClass({ String? name,  int? age}) {
    return MyClass._private(name:  name, age: age);
  }
}

void main() {
```

```
  var obj = MyClass(name: "Tarunika", age: 25);
  print("${obj.name}, ${obj.age}");
}
```