Mohammed Roshan Khan

Day 11 – Assignment

Inheritance in Dart:

Inheritance is a feature in Dart that allows a class (child/subclass) to inherit properties and methods from another class (parent/superclass), promoting code reuse and hierarchical classification.

Ex:-

```
class employee {
  String? name;
  int? id;
  String? designation;

  employee() {
    print("this a default constructor of super class");
  }

  void displayEmployee() {
    name = "roshan";
    id = 1;
    designation = "SE";

    print(
      "${this.name} - your id is ${this.id} and your designation is
${this.designation}",
    );
  }
}

class manager extends employee {
  manager() {
    print("this is the defalut constructor of the derived class");
  }

  void displayManager() {
    this.name = "suresh";
    this.id = 2;
```

```
    this.designation = "senior SE";

    print(
      "${this.name} - your id is ${this.id} and your designation is
${this.designation}",
    );
  }
}

void main() {
  manager m = manager();
  m.displayEmployee();
  m.displayManager();
}
```

Definition:

A **factory constructor** in Dart is a special type of constructor that **does not always create a new instance** of a class every time it's called. Instead, it can return an existing instance, modify data before creating an object, or even throw exceptions based on some logic.

Ex:-

```
// private constructor

class banking {
  String? accnumber;
  String? accholder;
  double? balance;

  banking._({this.accnumber, this.accholder, this.balance});

  // factory constructor

  factory banking.toCheck({
    String? accnumber,
    String? accholder,
    double? balance,
  }) {
    if (balance! < 100) {
```

```
      throw new Exception("Balance is insufficient");
    } else {
      return banking._(
        accnumber: accnumber,
        accholder: accholder,
        balance: balance,
      );
    }
  }

  double deposit(double amt) {
    return balance = balance! + amt;
  }

  double withDraw(double amt) {
    if (balance! > amt) {
      balance = balance! - amt;
    } else {
      throw new Exception("Less balance !");
    }

    return balance!;
  }
}

void main() {
  var b1 = banking.toCheck(
    accnumber: "a1",
    accholder: "Roshan",
    balance: 12500,
  );
  print(b1.deposit(5000));
  print("Balance is ${b1.withDraw(1000)}");
}
```

Abstract Class

An **abstract class** in Dart is a class that **cannot be instantiated directly**. It is designed to be a **base class** that other classes inherit from. It can contain both:

- **Abstract methods** (methods without a body) that **must be implemented** by subclasses.
- **Concrete methods** (methods with a body) that provide common functionality to subclasses.

```
// Abstract class
abstract class BankAccount {
  String accountNumber;
  double balance;

  BankAccount(this.accountNumber, this.balance);

  // Abstract method (no body)
  void withdraw(double amount);

  // Concrete method
  void deposit(double amount) {
    balance += amount;
    print("Deposited \$${amount}. New balance: \$${balance}");
  }

  void showBalance() {
    print("Account $accountNumber - Balance: \$${balance}");
  }
}

// Concrete class
class SavingsAccount extends BankAccount {
  double interestRate;

  SavingsAccount(String accountNumber, double balance, this.interestRate)
    : super(accountNumber, balance);

  @override
  void withdraw(double amount) {
    if (balance - amount < 0) {
      print("Insufficient funds in Savings Account.");
    } else {
      balance -= amount;
      print("Withdrawn \$${amount} from Savings Account.");
    }
  }
```

```
  void applyInterest() {
    double interest = balance * interestRate;
    deposit(interest);
    print("Interest of \$${interest} applied.");
  }
}

// Another concrete class
class CurrentAccount extends BankAccount {
  double overdraftLimit;

  CurrentAccount(String accountNumber, double balance, this.overdraftLimit)
    : super(accountNumber, balance);

  @override
  void withdraw(double amount) {
    if (balance + overdraftLimit - amount < 0) {
      print("Overdraft limit exceeded.");
    } else {
      balance -= amount;
      print("Withdrawn \$${amount} from Current Account.");
    }
  }
}

void main() {
  SavingsAccount sa = SavingsAccount("SA123", 1000.0, 0.05);
  sa.deposit(500);
  sa.withdraw(200);
  sa.applyInterest();
  sa.showBalance();

  print("----");

  CurrentAccount ca = CurrentAccount("CA456", 500.0, 1000.0);
  ca.withdraw(1200); // within overdraft
  ca.withdraw(500); // exceeds overdraft
  ca.showBalance();
}
```