

## 1. Dart-এ class সম্পর্কে ব্যাখ্যা

```
class ClassName {  
    <fields>  
    <getter/setter>  
    <constructor>  
    <functions>  
}
```

**Dart-এ ক্লাস (Class)** হলো একটি ক্লস্প্রিন্ট বা ছাঁচ, যার মাধ্যমে আপনি অবজেক্ট তৈরি করতে পারেন। ক্লাসে ভেরিয়েবল (**properties**) এবং মেথড (**functions**) থাকে যা একসাথে কোনো নির্দিষ্ট টাইপের অবজেক্টকে সংজ্ঞায়িত করে।

ক্লাসের বৈশিষ্ট্য

1. **Properties** বা ফিল্ডস: এগুলো হলো ভেরিয়েবল, যা অবজেক্টের ডেটা ধারণ করে।
2. **Methods** বা **Functions**: এগুলো হলো ফাংশন, যা ক্লাসের অবজেক্টের জন্য কার্যকর।
3. **Constructor**: একটি বিশেষ মেথড, যা অবজেক্ট তৈরি করার সময় চালিত হয়।

উদাহরণ ১: সাধারণ ক্লাস

dart

Copy code

```
class Person {  
    // Properties  
    String name;  
    int age;  
  
    // Constructor  
    Person(this.name, this.age);  
  
    // Method  
    void displayInfo() {  
        print('Name: $name, Age: $age');  
    }  
}  
  
void main() {  
    // Object creation  
    Person person1 = Person('Rahim', 25);
```

```
// Method call
person1.displayInfo();
}
```

আউটপুট:

yaml

Copy code

Name: Rahim, Age: 25

ব্যাখ্যা:

1. `class Person` একটি ক্লাস, যেখানে `name` এবং `age` হলো প্রপার্টি।
2. `Person(this.name, this.age)` হলো কন্সট্রাক্টর, যা অবজেক্ট তৈরি করার সময় প্রপার্টি সেট করে।
3. `displayInfo()` মেথড অবজেক্টের ডেটা প্রিন্ট করে।

উদাহরণ ২: ক্লাসের ফাংশনালিটিকে বাড়ানো

```
class Rectangle {
    // Properties
    double length;
    double width;

    // Constructor
    Rectangle(this.length, this.width);

    // Method to calculate area
    double calculateArea() {
        return length * width;
    }

    // Method to calculate perimeter
    double calculatePerimeter() {
        return 2 * (length + width);
    }
}

void main() {
    // Object creation
    Rectangle rect = Rectangle(10.5, 5.5);
}
```

```
print('Area: ${rect.calculateArea()}');  
print('Perimeter: ${rect.calculatePerimeter()}');  
}
```

## 2. Dart-এ Object সম্পর্কে ব্যাখ্যা

### Dart-এ অবজেক্ট (Object) নিয়ে বিস্তারিত ব্যাখ্যা

অবজেক্ট হলো একটি ক্লাসের ইনস্ট্যান্স। অর্থাৎ, ক্লাস হলো একটি ছাঁচ (**blueprint**), আর অবজেক্ট হলো সেই ছাঁচ থেকে তৈরি বাস্তব রূপ।

**Dart** একটি **Object-Oriented Programming (OOP)** ভাষা, যেখানে সবকিছুই অবজেক্ট।

---

অবজেক্ট কীভাবে কাজ করে?

1. ক্লাস থেকে তৈরি: অবজেক্ট ক্লাসের মাধ্যমে তৈরি হয়।
  2. **Properties** এবং **Methods**: ক্লাসে যেসব প্রপার্টি ও মেথড থাকে, অবজেক্ট সেই গুলো অ্যাক্সেস করতে পারে।
  3. ডেটা হোল্ড করে: অবজেক্ট রিয়েল-টাইম ডেটা ধারণ করে।
- 

উদাহরণ ১: সহজ উদাহরণ

dart

Copy code

```
class Car {  
  // Properties  
  String brand;  
  int year;  
  
  // Constructor  
  Car(this.brand, this.year);  
  
  // Method  
  void displayCarInfo() {  
    print('Brand: $brand, Year: $year');  
  }  
}
```

```
}

void main() {
    // Object creation
    Car myCar = Car('Toyota', 2021);

    // Accessing properties and methods
    myCar.displayCarInfo();
}
```

আউটপুট:

yaml

Copy code

Brand: Toyota, Year: 2021

ব্যাখ্যা:

1. **Car** ক্লাসে **brand** এবং **year** নামের দুটি প্রপার্টি আছে।
  2. কন্সট্রাক্টর দিয়ে প্রপার্টিগুলোর মান নির্ধারণ করা হয়েছে।
  3. **myCar** অবজেক্ট তৈরি করা হয়েছে এবং **displayCarInfo()** মেথড কল করে ডেটা প্রিন্ট করা হয়েছে।
- 

উদাহরণ ২: মাল্টিপল অবজেক্ট

dart

Copy code

```
class Student {
    String name;
    int roll;

    Student(this.name, this.roll);

    void showDetails() {
        print('Name: $name, Roll: $roll');
    }
}
```

```
void main() {  
    // Creating multiple objects  
    Student student1 = Student('Rahim', 101);  
    Student student2 = Student('Karim', 102);  
  
    // Calling methods  
    student1.showDetails();  
    student2.showDetails();  
}
```

আউটপুট:

yaml

Copy code

Name: Rahim, Roll: 101

Name: Karim, Roll: 102

ব্যাখ্যা:

1. দুইটি আলাদা অবজেক্ট তৈরি হয়েছে **student1** এবং **student2**।
2. প্রতিটি অবজেক্ট আলাদা ডেটা ধারণ করে।

---

উদাহরণ ৩: অবজেক্টের মাধ্যমে অ্যাক্সেস এবং মডিফাই

dart

Copy code

```
class BankAccount {  
    String accountHolder;  
    double balance;  
  
    BankAccount(this.accountHolder, this.balance);  
  
    void deposit(double amount) {  
        balance += amount;  
        print('Deposited: $amount, New Balance: $balance');  
    }  
  
    void withdraw(double amount) {
```

```

        if (amount <= balance) {
            balance -= amount;
            print('Withdrawn: $amount, Remaining Balance:
$balance');
        } else {
            print('Insufficient Balance!');
        }
    }
}

void main() {
    // Object creation
    BankAccount account = BankAccount('Alice', 5000.0);

    // Accessing methods
    account.deposit(2000.0);
    account.withdraw(3000.0);
    account.withdraw(5000.0);
}

```

আউটপুট:

yaml

Copy code

```

Deposited: 2000.0, New Balance: 7000.0
Withdrawn: 3000.0, Remaining Balance: 4000.0
Insufficient Balance!

```

ব্যাখ্যা:

3. **BankAccount** ক্লাস একটি ব্যাংক অ্যাকাউন্টকে মডেল করে।
4. অবজেক্ট **account** প্রপার্টি মডিফাই এবং মেথড কল করতে পারে।
5. অবজেক্ট বাস্তব ডেটা ম্যানেজ করতে পারে, যেমন জমা বা উত্তোলন।

6. Dart-এ **Inheritance** সম্পর্কে ব্যাখ্যা
7. Dart-এ **Polymorphism** সম্পর্কে ব্যাখ্যা
8. Dart-এ **Interfaces** সম্পর্কে ব্যাখ্যা
9. Dart-এ **Abstract class** সম্পর্কে ব্যাখ্যা

