**Dart – Day 3**

### **Functional Programming in Dart**

Functional programming focuses on writing code using functions as first-class citizens - meaning functions can be assigned to variables, passed as arguments, and returned from other functions.

### **Example 1: Assigning function to variable**

void main()

{  
 var greet = (String name) => "Hello, $name"; // anonymous function  
 print(greet("Chandini")); // Hello, Chandini  
}

### **Example 2: Passing function as argument**

void sayHello()

{

print("Hello!");

}

void callFunction(Function func)

{

func(); // call the function passed

}

void main()

{

callFunction(sayHello); // passing function as argument

}

### **Example 3: Returning function (closure)**

Function multiplier(int factor)

{  
 return (int number) => number \* factor;  
}  
  
void main()

{  
 var doubleIt = multiplier(2);  
 print(doubleIt(6)); // 12  
}

* **Data Types**

### **1. int**

In Dart, int is used for whole numbers (positive or negative) without decimals.

**Example:**

void main() {  
 int age = 25;  
 int year = 2025;  
 print(age); // 25  
 print(year); // 2025  
}

### **2. BigInt**

BigInt is used when numbers are too large to fit in a normal int.

**Example:**

void main() {  
 BigInt big = BigInt.parse("383657758108925447369389475973485");  
 print(big);   
}

### **3. double**

In Dart, double is used for decimal numbers (floating-point values).

**Example:**

void main() {  
 double pi = 3.14159;  
 double price = 99.99;  
 print(pi); // 3.14159  
 print(price); // 99.99  
}

### **4. num**

num is a supertype of both int and double. It can store either whole numbers or decimals.

**Example:**

void main() {  
 num x = 10; // int  
 num y = 5.5; // double  
 print(x); // 10  
 print(y); // 5.5  
}

### **Records**

A record is a lightweight data type that can hold multiple values without creating a class.

**Example:**

void main() {  
 var employee = ("Chandini", 22, isFullTime: true);  
  
 print(employee.$1); // Chandini (positional field)  
 print(employee.$2); // 22  
 print(employee.isFullTime); // true (named field)  
}

### **Returning Multiple Values**

Dart functions can return multiple values using records.

**Example:**

(String, int) getUser() {  
 return ("Chandini", 22);  
}  
  
void main() {  
 var user = getUser();  
 print(user.$1); // Chandini  
 print(user.$2); // 22  
}

**Another Example**

main()

{

var myData=("Chandini","L&D","I-Exceed",yoe:0);

print(myData.runtimeType);

print('Employee name is ${myData.$1}');

print(myData.$3);

print(myData.yoe);

var args= getQualification();

print(args.$1);

print(args.cgpa);

}

(String,{double cgpa}) getQualification()

{

return ("B.Tech",cgpa:8.6);

}

**Output**

(String, String, String, {int yoe})

Employee name is Chandini

I-Exceed

0

B.Tech

8.6