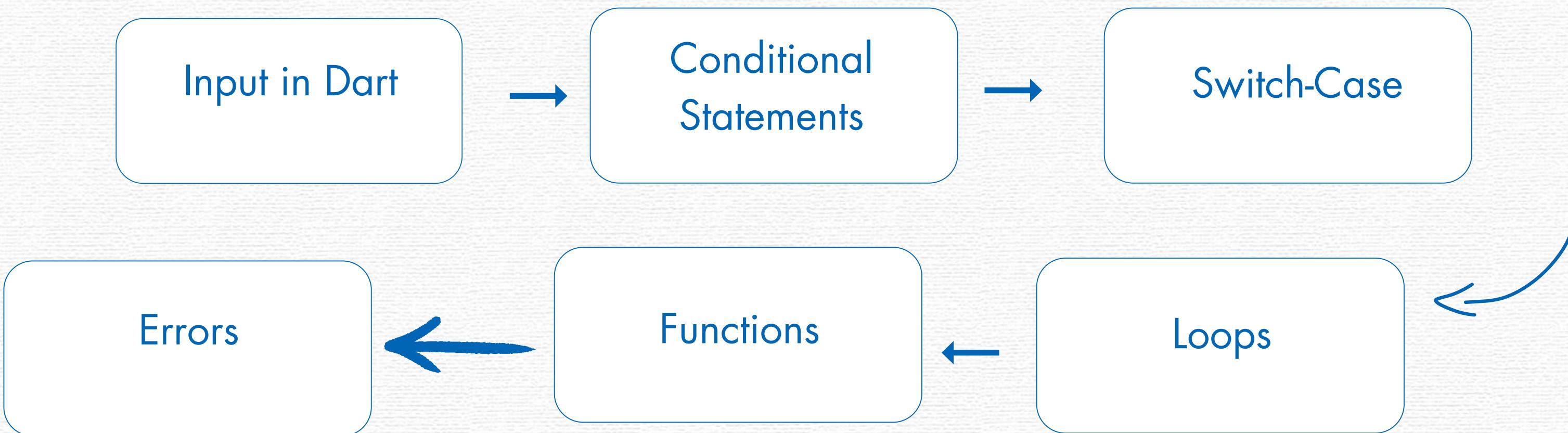


MOTOR

SESSION 2

#create_share_innovate

L i p s i l l i n g i n i





FCI - Helwan
Student Branch



PROJECT

Presentation
15-mins

INPUT

Using `dart:io` Package

`stdin.readLineSync()` :
returns a nullable String?.

```
import 'dart:io';

void main() {
  print("Enter your name:");
  String? name = stdin.readLineSync();
  print("Hello, $name!");
}
```

NUNJIRI

needing **parse** to convert the input to another type

```
import 'dart:io';

void main() {
  print("Enter your age:");
  int age = int.parse(stdin.readLineSync()!);
  print("You are $age years old.");
}
```

CONTINUOUS LEARNING

if Statement:

Checks a condition and executes code if true.

```
if (age > 18) {  
    print('Adult');  
}
```

UNILINK iMEN

else Statement:

Executes if the if condition is false.

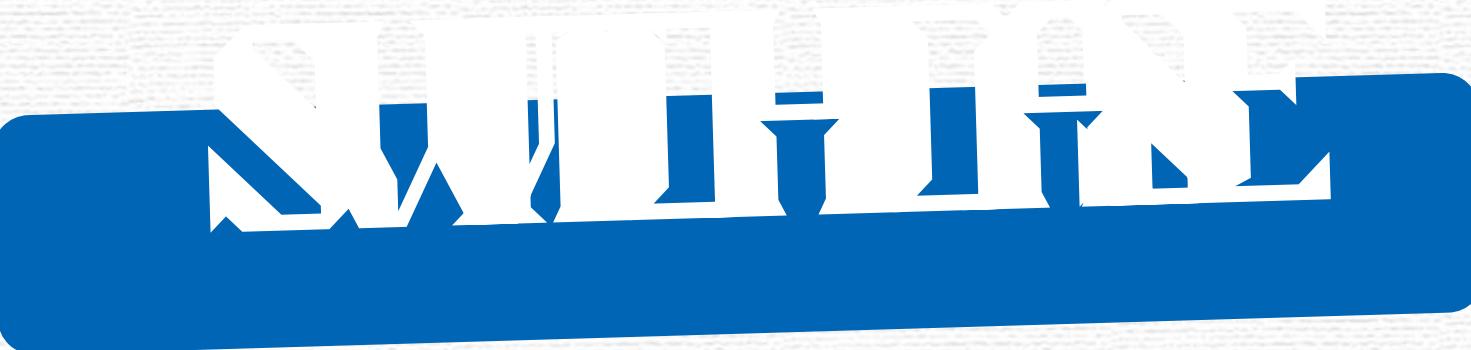
```
if (age > 18) {  
    print('Adult');  
} else {  
    print('Minor');  
}
```

UNIVERSITY LEVEL

else if Statement:

Checks multiple conditions

```
if (score >= 90) {  
    print('A');  
} else if (score >= 80) {  
    print('B');  
} else {  
    print('C');  
}
```

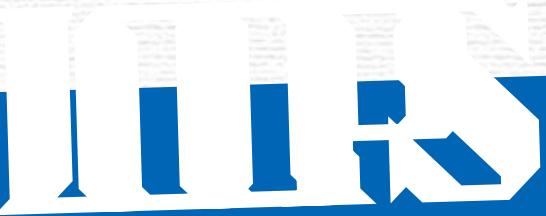


Efficient for checking a single variable against multiple values

Note

`break` is mandatory to prevent fall-through.
`default` executes if no cases match.

```
var grade = 'A';
switch (grade) {
    case 'A':
        print('Excellent');
        break;
    case 'B':
        print('Good');
        break;
    default:
        print('Try again');
}
```



for Loop:

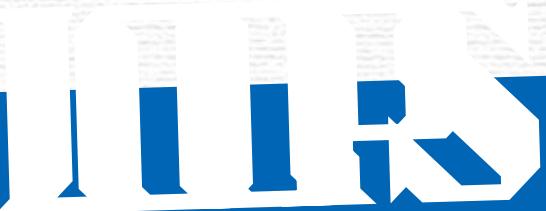
Runs a fixed number of times.

```
for (int i = 0; i < 5; i++) {  
    print(i);  
}
```

while Loop:

Runs as long as a condition is true.

```
int i = 0;
while (i < 5) {
    print(i);
    i++;
}
```



do-while Loop

Executes at least once, then checks the condition.

```
int i = 0;
do {
    print(i);
    i++;
} while (i < 5);
```

QUESTION

Write a program to check whether a number is prime using a for loop .



FUNCTIONS

A function is a block of code that performs a specific task.

Function Definition :

```
int add(int a, int b) {  
    return a + b;  
}
```

Function Calling:

```
void main() {  
    var int result = add(a: 5, b: 3);  
    print(object: result); // Output: 8  
}
```



FUNCTIONS

Return Type:

Functions can return values. In the Previous example, the return type is int.
If a function doesn't return a value, use **void**.

```
void printMessage(String message) {  
    print(object: message);  
}
```

Function Parameters

Positional Parameters:

Parameters are passed in the order they are defined in the function.

```
void greet(String name, int age) {  
    print(object: 'Hello, $name! You are $age years old.');//  
}
```

Optional Parameters

Optional Parameters

Parameters can be optional. Use [] for optional parameters.

```
void greet(String name, [int age = 18]) {  
  print(object: 'Hello, $name! You are $age years old.');//  
}  
  
Run | Debug  
void main() {  
  greet(name: 'Alice');           // Uses default age of 18  
  greet(name: 'Bob', age: 25);    // Overrides default age  
}
```

GENERIC FUNCTIONS

:: CAUTION ::

in Dart, positional optional parameters ***cannot be the first parameter.***

```
// ✗ This will cause a syntax error!
void greet([String name, int age]) {
    print(object: 'Hello, $name! You are $age years old.');
}
```

Function Parameters

Named Required Parameters:

Named parameters allow for more flexible function calls. Use {} to define them.

```
void greet({required String name, required int age}) {  
  print(object: 'Hello, $name! You are $age years old.');//  
}  
  
dynamic greet(dynamic name: 'Bob', dynamic age: 30); // Named parameter usage
```

Function Parameters

Named optional Parameters (BY DEFUALT):

Named parameters allow for more flexible function calls. Use {} to define them.

```
void greet({String? name, int? age}) {  
  print(object: 'Hello, $name! You are $age years old.');//  
}  
  
dynamic greet(dynamic name: 'Charlie'); // Optional parameter usage
```



Function Parameters

Named optional Parameters (BY DEFUALT):

Named parameters allow for more flexible function calls. Use {} to define them.

```
void greet({String? name, int? age}) {  
  print(object: 'Hello, $name! You are $age years old.');//  
}  
  
dynamic greet(dynamic name: 'Charlie'); // Optional parameter usage
```



FUNCTIONS

An anonymous function (or **lambda function**) is a function without a name. It's mainly used when you need a short function for a specific task and don't want to define a separate named function.



FUNCTIONS IN

```
// Anonymous function assigned to a variable
var multiply = (int a, int b) {
    return a * b;
};

Run | Debug
void main() {
    print(multiply(3, 4)); // Output: 12
}
```

Explanation:

- We assigned an unnamed function to the variable `multiply`.
- It takes two parameters and returns their product.
- We call it like a regular function: `multiply(3, 4)`.



Syntax Errors:

Occur due to incorrect syntax.

Example: Missing semicolon.

Fix: Correct the syntax.

Runtime Errors:

Occur during program execution.

Example: Division by zero.

Fix: Add validation or exception handling.

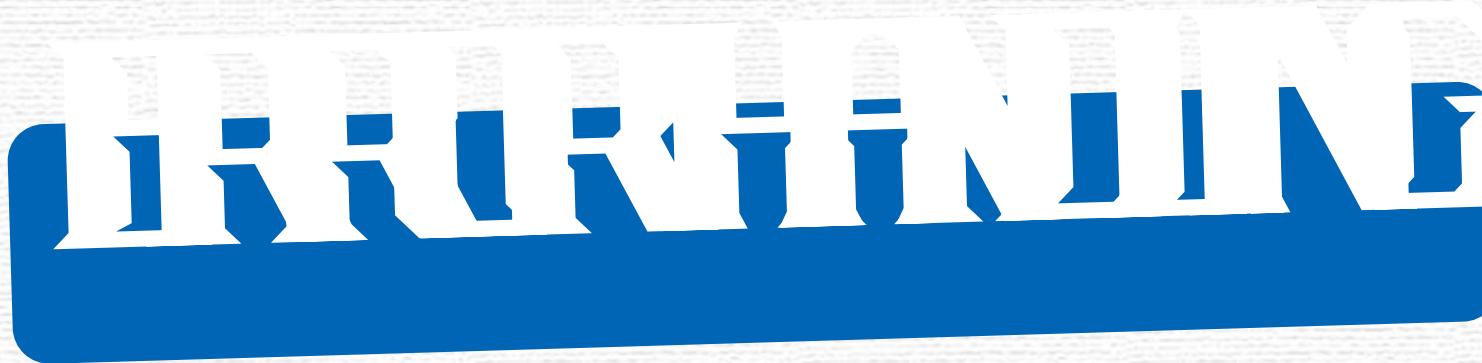


Logical Errors:

Incorrect program logic leading to wrong results.

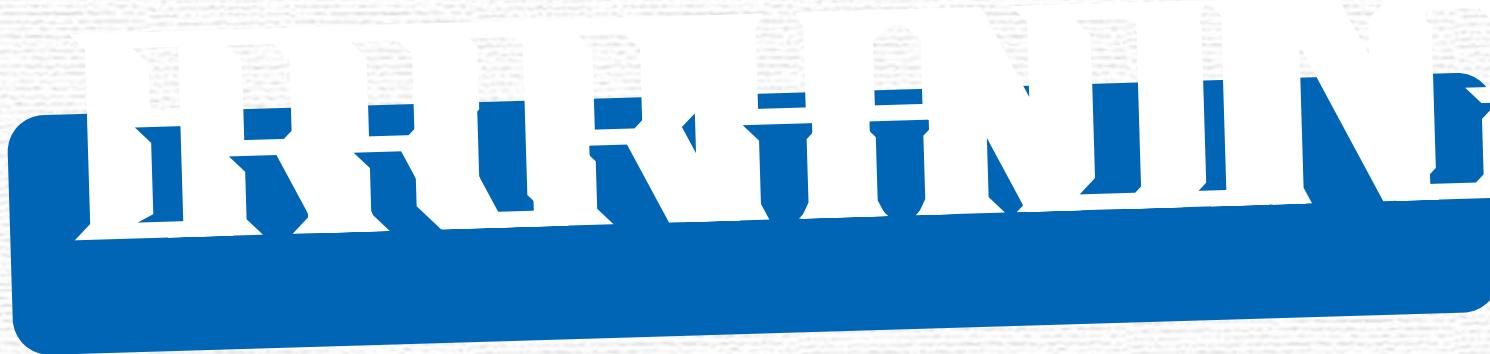
Example: Using + instead of -.

Fix: Review and correct the logic.



try-catch-finally:

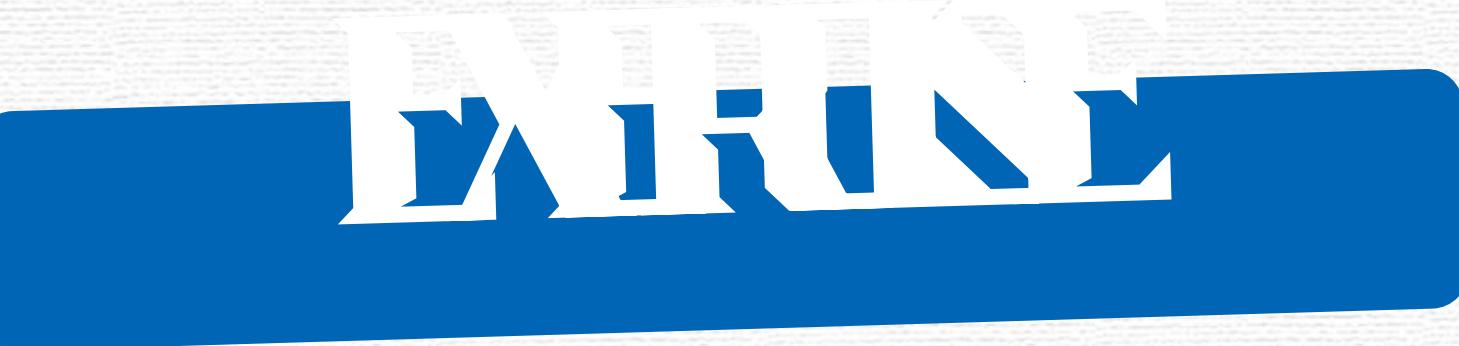
- **try:** Wraps code that might throw an exception.
- **catch:** Catches and handles the exception.
- **finally:** Executes regardless of an error occurring.



```
try {  
    int result = 10 ~/ 0;  
} catch (e) {  
    print('Error: $e');  
} finally {  
    print('Execution completed');  
}
```

Note:

For the best implementation, use try-catch for handling user and platform faults; otherwise, use if-else statements.



Write a function `calculateDeliveryCost` that calculates the delivery cost based on:

Required parameters:

weight (kg)

distance (km)

Optional parameters:

`isExpress` (default: false) → Increases cost by 20% if true.

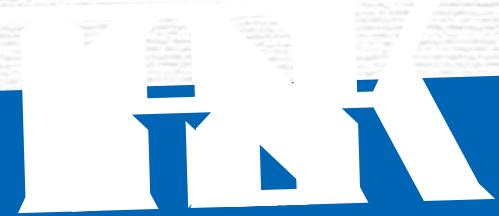
`discount` (optional) → Reduces cost by the given percentage.

Cost Calculation:

$$\text{cost} = (\text{weight} \times 5) + (\text{distance} \times 2)$$

If express delivery is selected, increase cost by 20%.

If a discount is given, reduce cost accordingly.



Build a simple calculator program that supports basic operations (addition, subtraction, multiplication, division).



IEEE | FCI - Helwan
Student Branch



FCI HELWAN
NEW TIME

S25' Training Sessions