Lab 3: Dart Programming Fundamentals 1

Coding Exercises

School of Computing
Department of Computer Science
New Uzbekistan University in Tashkent

September 8, 2025

Contents

1 The `main` Function **3**

2 Variables **4**

3 Control Flow **5**

4 Functions **6**

5 Comments **7**

1 The `main` Function

The `main()` function is the starting point of every Dart application. The code inside its curly braces {} is what runs when you execute the program.

Task 1 (Sample with Solution)

- **Goal:** Write a program that prints the message "Hello, World!" to the console.

Solution:

```dart
// The main function is the entry point for all Dart programs.
void main() {
  // The print() function displays text in the console.
  print('Hello, World!');
}
```

Your Tasks

Instructions: For each task below, write your solution in a Dart code block.

Task 2

Write a program that prints your full name to the console.

Task 3

Write a program that prints two separate lines of text:
Line 1: `Welcome to Dart Programming.`
Line 2: `Let's start coding!`

Task 4

Using a single `print()` statement, write a program that prints a multi-line address for a fictional location in Tashkent. Example:

```
1 Amir Temur Avenue
Tashkent, Uzbekistan
100000
```

Task 5

Write a program that prints the following message, including the current year (2025): `The current year is 2025.`

Task 6

Write a program that prints a welcome message to your university. For example: `Welcome, students of INHA University in Tashkent!`

2 Variables

Variables are containers for storing data values. In Dart, you can declare variables using keywords like `String`, `int`, `double`, `bool`, and `var`.

Task 1 (Sample with Solution)

- **Goal:** Declare an integer variable called `age`, assign it the value `20`, and print a message including the age.

Solution:

```
void main() {
  // Declare an integer variable 'age' and initialize it with 20.
  int age = 20;

  // Use string interpolation ($) to include the variable's value in the output.
  print('The student is $age years old.');
}
```

Your Tasks

Task 2

Declare a `String` variable named `favoriteCity`, assign it the name of your favorite city, and print the sentence: `My favorite city is [Your City Name].`

Task 3

Declare a `double` variable named `bookPrice` and assign it a value of `15.99`. Print the message: `The price of the book is $bookPrice USD.`

Task 4

Declare a boolean variable named `isLearningDart` and set its value to `true`. Print a message that says: `Am I learning Dart? The answer is [true/false].`

Task 5

Declare two variables: a `String` named `weather` with the value `"Sunny"` and an `int` named `temperature` with the value `32`. Print them in a single sentence: `Today's weather is Sunny and the temperature is 32 degrees Celsius.`

Task 6

Declare a variable using the `var` keyword, name it `numberOfApples`, and assign it the value `10`. Print its value. On the next line, change the value of `numberOfApples` to `12` and print its new value.

3 Control Flow

Control flow statements (like `if`, `else`, and `for` loops) allow you to control the order in which your code is executed based on certain conditions.

Task 1 (Sample with Solution)

- **Goal:** Check if a number is greater than 10.

Solution:

```
void main() {
  int number = 15;

  // The 'if' statement checks if the condition (number > 10) is true.
  if (number > 10) {
    print('The number is greater than 10.');
  } else {
    // This block runs if the condition is false.
    print('The number is not greater than 10.');
  }
}
```

Your Tasks

Task 2

Declare an integer variable `studentScore` and set it to `75`. Write an `if-else` statement that prints "Pass" if the score is 60 or greater, and "Fail" otherwise.

Task 3

Declare an integer variable `hour` and set it to `14`. Write an `if-else if-else` statement that prints:

- "Good morning" if `hour` is less than 12.
- "Good afternoon" if `hour` is between 12 and 18 (inclusive).
- "Good evening" for any other time.

Task 4

Write a `for` loop that prints all the numbers from 1 to 10, each on a new line.

Task 5

Write a `for` loop that counts down from 5 to 1 and then prints "Liftoff!" after the loop is finished.

Task 6

Write a `for` loop that iterates from 1 to 20. Inside the loop, use an `if` statement to print only the numbers that are multiples of 3.

4 Functions

Functions are blocks of code that perform a specific task. They help you organize your code, make it reusable, and easier to read.

Task 1 (Sample with Solution)

- **Goal:** Create a function that prints a standard greeting and call it from `main`.

Solution:

```
// This function prints a greeting. It doesn't take any inputs or return any
    values.
void showGreeting() {
  print('Welcome to our program!');
}

void main() {
  // We call the function here to execute its code.
  showGreeting();
}
```

Your Tasks

Task 2

Create a function named `greetByName` that accepts one `String` argument called `name`. The function should print `Hello, [name]!`. From `main`, call this function with your name.

Task 3

Create a function named `multiply` that takes two `int` arguments, `a` and `b`, and `returns` their product. In `main`, call this function with the numbers 7 and 6, and print the returned result.

Task 4

Create a function named `isPositive` that takes one `int` argument `number`. The function should return a `bool`: `true` if the number is greater than 0, and `false` otherwise. In `main`, test this function with a positive and a negative number and print the results.

Task 5

Create a function named `calculateAverage` that takes two `double` arguments, `num1` and `num2`, and returns their average. Call this function from `main` with values `10.5` and `20.5` and print the result.

Task 6

Rewrite the `multiply` function from Task 3 using Dart's arrow syntax (`=>`). The function should still take two integers and return their product.

5 Comments

Comments are notes in your code that are ignored by the compiler. They are used to explain what your code does, making it easier for you and others to understand.

Task 1 (Sample with Solution)

- **Goal:** Write a simple program and add both a single-line and a multi-line comment.

Solution:

```
/*
  Author: Gemini
  Date: September 8, 2025
  This program demonstrates how to use comments in Dart.
*/

void main() {
  // This is a single-line comment. It explains the next line of code.
  String message = 'Comments make code readable.';

  print(message);
}
```

Your Tasks

Task 2

Copy the code below. Add a single-line comment above the `planet` variable declaration that explains what the variable stores.

```
void main() {
  String planet = 'Earth';
  print('We live on planet $planet.');
}
```

Task 3

Copy the code below. Add a multi-line comment at the very top of the program. The comment should include your name, the current date, and a brief description: "This is my first Dart function."

```
void sayHello() {
  print('Hello from a function!');
}

void main() {
  sayHello();
}
```

Task 4

The code below prints two messages. Use comments to "comment out" the first `print` statement so that only the second message is displayed when the program runs.

```
void main() {
  print('This message should not appear.');
  print('This message should appear.');
}
```

Task 5

Add a documentation comment (`///`) to the `calculateCircleArea` function below. The comment should explain that the function calculates the area of a circle given its radius.

```
import 'dart:math';

// Your documentation comment goes here
double calculateCircleArea(double radius) {
  return pi * radius * radius;
}

void main() {
  print(calculateCircleArea(5.0));
}
```

Task 6

The code below has a syntax error because of a broken comment. Find the error, fix it, and add a correct single-line comment explaining what the `speedOfLight` variable represents.

```
void main() {
  /* This variable stores a very important universal constant.
  int speedOfLight = 299792458; // in meters per second
  print(speedOfLight);
```