BCA Optimized Notes by Yash

Semester IV - Mobile Programming

Table of Contents

Unit 1 - Chapter 1

Mobile Programming

Mobile programming is software development for mobile devices.

Types

There are two types of mobile programming.

Native Programming - Single Platform	Hybrid Programming - Multiple Platforms
Faster and more reliable	Usually slower
Different languages for different platforms	Single language for all platforms
Cost of development is high	Cost of development is low
Allows usage of device-specific functionalities	Limited usage of device-specific functionalities

Flutter

- Flutter is an open-source UI framework developed by Google that allows developers to build native-quality mobile, web, and desktop applications from a single codebase.
- Flutter is a valuable modern tool used to create stunning cross-platform applications that render native code on each device and OS.
- Flutter is compatible with Android, iOS, Linux, macOs, Windows, etc.

Key Components

- **SDK (Software Development Kit)**
  - A set of instruments that serve as aid in development of applications.
  - Includes tools for converting code to native machine code.
  - Works on code for iOS and Android.
- **Framework (UI Library using Widgets)**
  - A set of reusable user interface components.
  - They can be customized according to the developer's purposes.
  - Components include buttons, text inputs, slides, etc.

Dart

- Dart is an open-source general-purpose programming language developed by Google. It supports application development in client and server sides.
- It is widely used for the development of Android apps, iOS apps, IoT, and web applications using the Flutter Framework.
- Dart is an object-oriented language and has a syntax quite similar to that of Java.
- Dart is extensively used to create single-page websites and web-applications.

Unit 1 - Chapter 2

Basics of Dart

Keywords

assert	default	finally	rethrow	try
break	do	for	return	var
case	else	if	super	void
catch	enum	in	switch	while
class	extends	is	this	with
const	false	new	throw	
continue	final	null	true	

Data Types

Types	Keywords
Numbers	int, double
Strings	string
Booleans	bool
Records	((value1, value2))
Lists	(list)
Sets	(set)
Maps	(map -> key value paired data)
Null	null

Operators

Types	Operators
Arithmetic	`+ - -expr * / ~/ %`
Relational	`< <= == > >= !=`
Type Test	`as is is!`
Bitwise	`& ^ |`
Assignment	`= ??= += -= *= /= %=`
Logical	`&& || ??`
Conditional	`expr1 ? expr2: expr3`

Collection Types

Lists

- Lists are similar to arrays in other programming languages.
- They are used to represent a collection of objects, or an ordered group of objects.
- The core libraries in Dart are responsible for the list class and its manipulation.
- There are two types of Lists:
    - Fixed Length List
    - Growth List

Sets

- Sets are a special case of Lists where all inputs are unique.
- They can also be interpreted as an unordered array with unique values.
- Sets come in play when we want to store unique values in a single variable without considering the order of the inputs.
- Sets are declared by the use of the "set" keyword.

Records

- Records are an anonymous, inmutable, and aggregate collection type.
- Unlike other collections, records are fixed size, heterogeneous, and typed.
- Records are real values: you can store variables, nest them, pass them to and from functions, and store them in data structures.

Maps

- Maps are a dictionary-like collection type that have key-value pairs.
- There is no restriction on the type of data that goes in the map collection type.
- Maps are very flexible and can move their size based on requirements.
- It is important to note that all keys need to be unique.

Functions

- Functions are a set of statements that take inputs, do specific computations, and provide outputs.
- Functions are created when certain statements are repeatedly occuring in the program and a function is created to replace them.
- Functions make it easy to divide a complex program into smaller subprograms and increase the code reusability of the program.