SLIIT UNI
THE KNOWLEDGE UNIVERSITY

Mobile Application Development

Introduction to App Development

SLIIT UNI
THE KNOWLEDGE UNIVERSITY

Introduction to the module

- Student Centered Learning module
- Module Code: IT2010
- Credit Value: 4
- Method of Delivery
  - Lectures – 1 hour/week
  - Tutorials (practical) – 2 hours/week
  - Labs - 2 hours/week
- Courseweb Enrollment Key: IT2010

SLIIT UNI
THE KNOWLEDGE UNIVERSITY

Lecture Plan

- Introduction to App Development
- Mobile Platforms and Application Development fundamentals
- Mobile Interface Design Concepts and UI/UX Design
- Introduction to Android Operating System
- Main Components of Android Application
- Sensors and Media Handling in Android Applications
- Data Handling in Android Applications
- Android Application Testing and security aspects

Assessment Criteria

Assessment	Weight (%)
Midterm (MCQ)	20
LAB Exam 1	10
LAB Exam 2	10
LAB Exam 3	10
Final Exam	50

Lecturer Panel

- Lecturer in-charge of the module
  - Mr. Thusithanjana Thilakarathna (thusithanjana.t@sliit.lk)

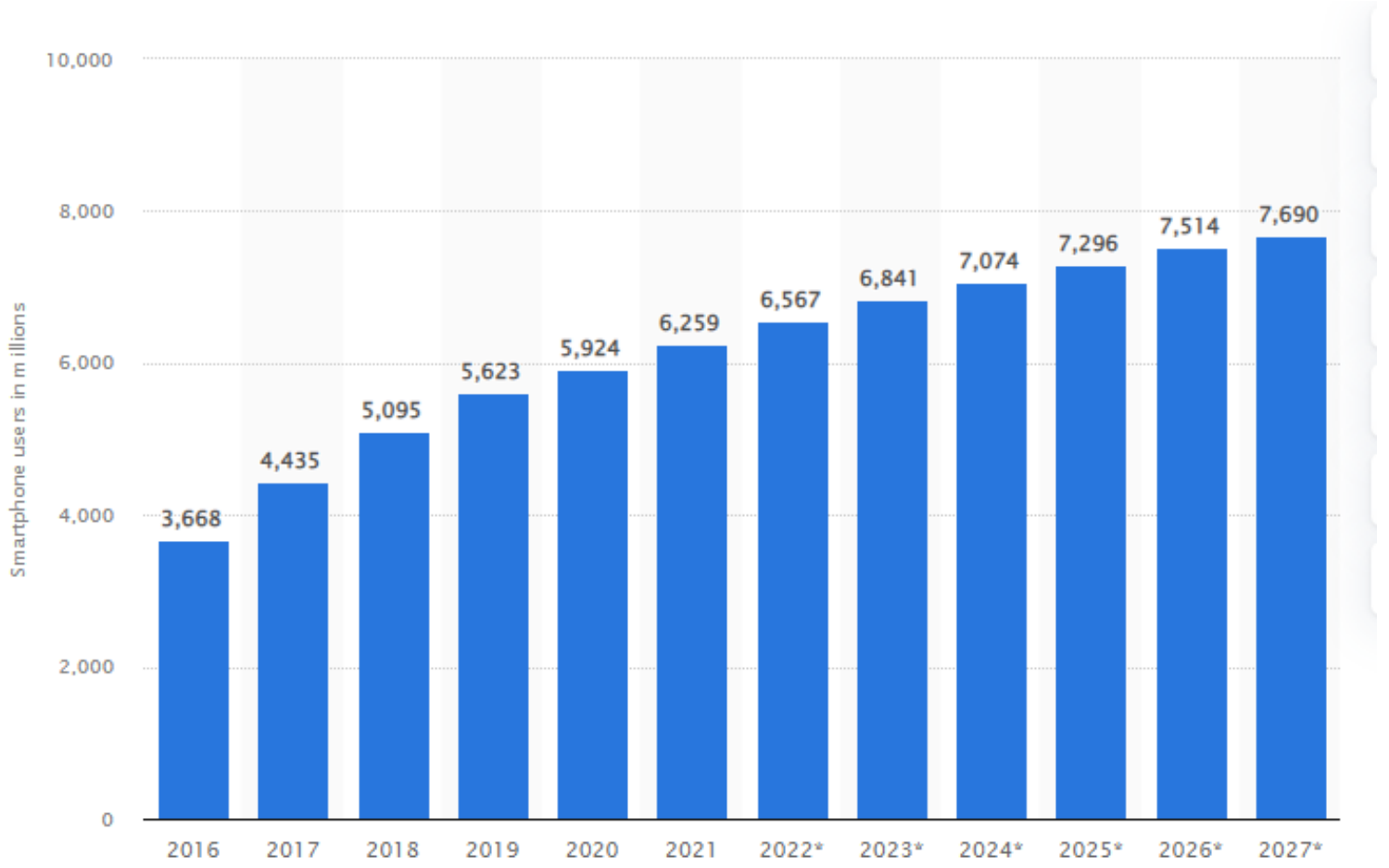
Learning Outcomes of the Lecture

- At the end of this Lecture students will be able to:
  - Understand the fundamentals of mobile Application Development
  - Explain different mobile platforms and related technologies.
  - Code a simple program in Kotlin

Why a Mobile App?

- Mobile phones are no longer the ordinary communication device. It has various incredible features and opportunities offered to the users.
- The number of smartphone users is growing up day by day. (In 2020, it's 3.5 billions)
- Business organizations are more like to have mobile applications for their business instead of investing in a mobile friendly version of their websites.
- Good mobile application will add value to the business.

Smartphone users' growth around the world



Reference: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/

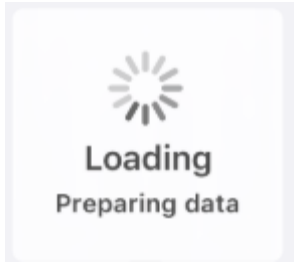
Mobile Application Development

- Mobile application development refers to the creation of apps for use on devices such as tablets, smartphones, automobiles and watches.
- Mobile development often incorporates features of mobile devices that may not be available on desktop devices.

  An example of this is the ability to operate a device or play a game simply by moving the smartphone around in space.

Key Features of Mobile Applications

- Great UI (User Interface)
- Fast loading time and high performance
- Extremely helpful user support
- Adapts to a user's needs
- Compatible with a mobile platform



Reasons for Mobile App failures

- The app doesn't have a market
- The app does not have adequate security
- The app does not perform quickly enough
- The app does not fully consider UX/UI
- The app's listing in the marketplace is not persuasive
- Hard to adjust web version to the smartphone screen
- Due to limited functions

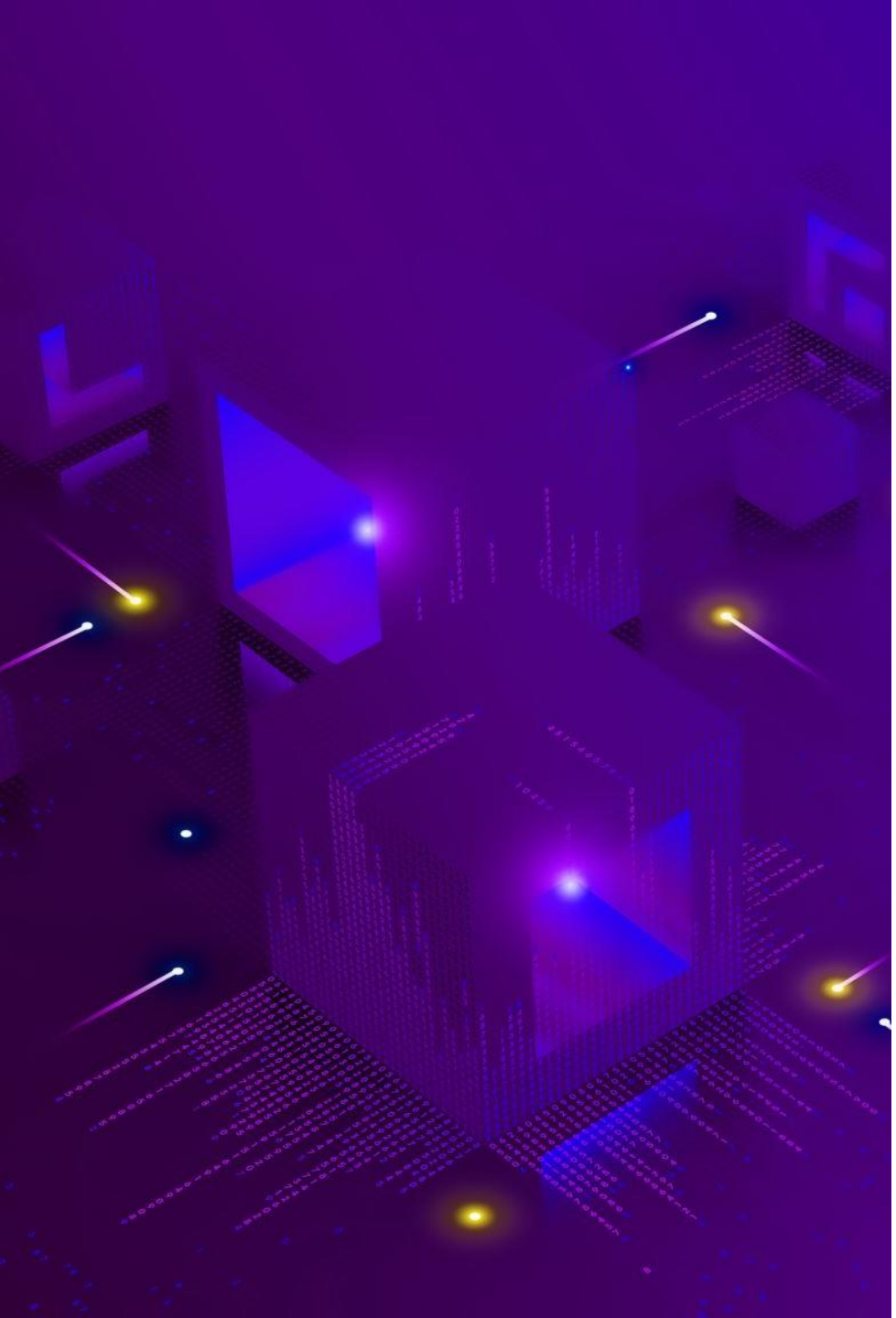
Mobile app development platforms

Android – Android Studio

iOS – Xcode

HarmonyOS – DevEco Studio

Cross Platform – Flutter, React Native, Xamarin

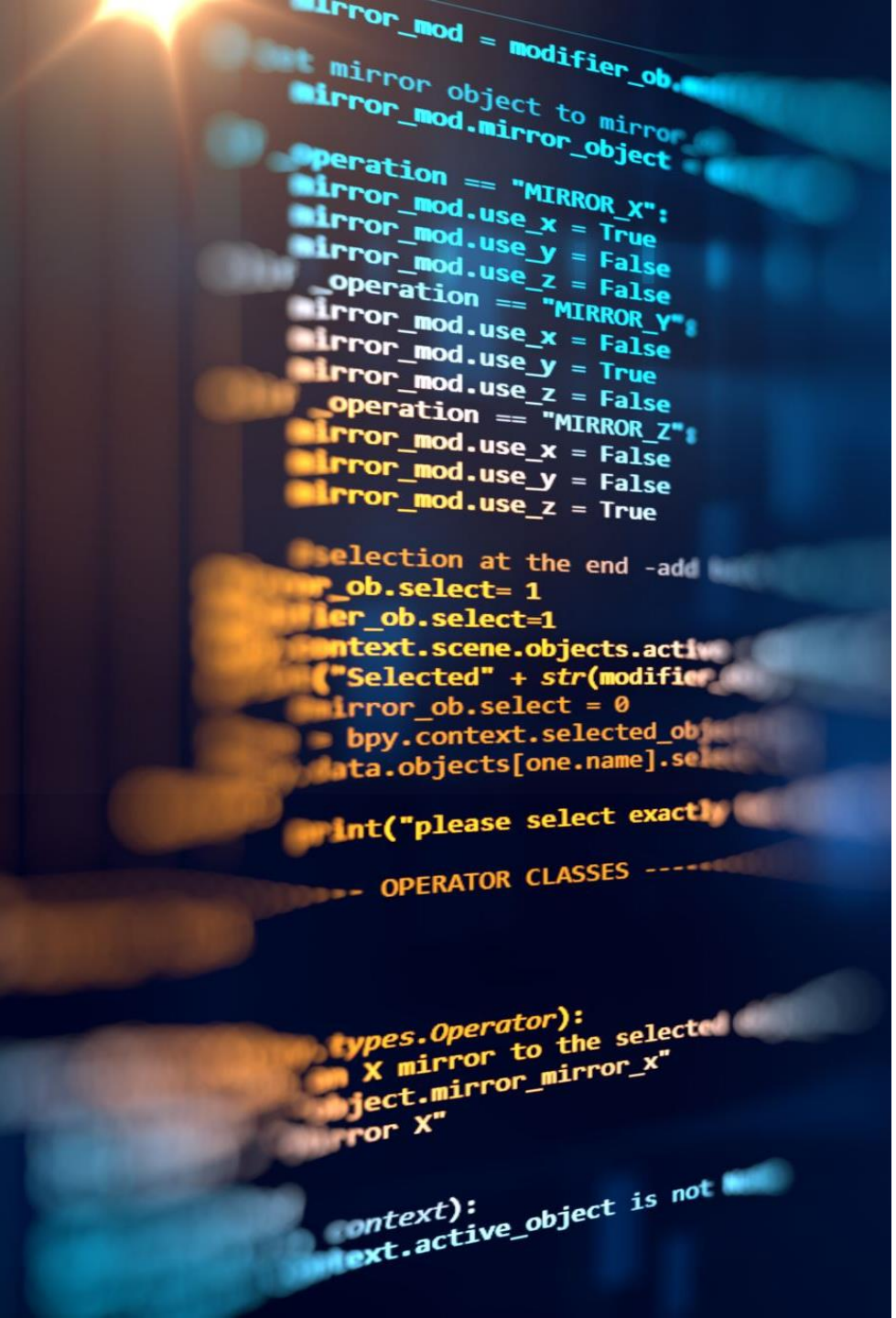


Android Studio

- Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.
- It is a replacement for the Eclipse Android Development Tools (E-ADT) as the primary IDE for native Android application development.
- Android Studio was announced on May 16, 2013, at the Google I/O conference.
- At the end of 2015, Google dropped support for Eclipse ADT, making Android Studio the only officially supported IDE for Android development.
- On May 7, 2019, Kotlin replaced Java as Google's preferred language for Android app development. Java is still supported, as is C++.

Kotlin

- Modern, concise and safe programming language
- Easy to pick up, so you can create powerful applications immediately.
- https://kotlinlang.org/



SLIIT UNI
THE KNOWLEDGE UNIVERSITY

Why is Android development Kotlin-first?

- **Expressive and concise:** You can do more with less. Express your ideas and reduce the amount of boilerplate code. 67% of professional developers who use Kotlin say Kotlin has increased their productivity.
- **Safer code:** Kotlin has many language features to help you avoid common programming mistakes such as null pointer exceptions. Android apps that contain Kotlin code are 20% less likely to crash.
- **Interoperable:** Call Java-based code from Kotlin or call Kotlin from Java-based code. Kotlin is 100% interoperable with the Java programming language, so you can have as little or as much of Kotlin in your project as you want.
- **Structured Concurrency:** Kotlin coroutines make asynchronous code as easy to work with as blocking code. Coroutines dramatically simplify background task management for everything from network calls to accessing local data.

Things you should know about Kotlin

- Variable Declaration
- Type inference
- Null Safety
- Conditionals
- Functions
- Classes
- Asynchronous code

Kotlin: Variable Declaration

- Use ‘val’ for a variable whose value never changes. You can't reassign a value to a variable that was declared using val.
- Use var for a variable whose value can change.

```
var count: Int = 10
```

```
var count: Int = 10
count = 15
```

Type Inference

- the Kotlin compiler can infer the type based on the type of the assigned value.

```
val name = "James Bond"
val age = 50
val isFlag = true
val pi = 3.14f
```

Null Safety

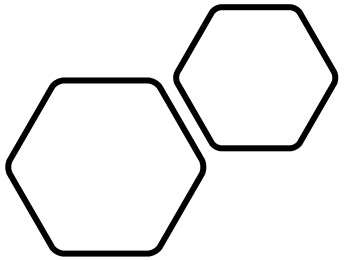
- In some languages, a reference type variable can be declared without providing an initial explicit value.
- In these cases, the variables usually contain a null value.
- Kotlin variables can't hold null values by default. This means that the following snippet is invalid:

```
// Fails to compile
val languageName: String = null
```

- For a variable to hold a null value, it must be of a nullable type.
- You can specify a variable as being nullable by suffixing its type with ?, as shown in the following example:

```
val languageName: String? = null
```

Thank you!



The tutorial sessions will include more Kotlin programming exercises.