

UNIT 1

Introduction to Mobile and Mobile Programming

(2 Hours) — Mobile Programming (CACS351)

1.1 Mobile Device

A **mobile device** is a portable, battery-powered computing device that can connect to wireless networks and be carried and used while the user is on the move. Smartphones, tablets, smartwatches, e-readers and handheld game consoles are all mobile devices. Modern mobile devices combine computing, communication, sensing and multimedia capabilities in a single pocket-sized unit.

Features of Mobile Devices

- **Portability** – small size and light weight, designed for one-hand or two-hand use.
- **Wireless connectivity** – Wi-Fi, cellular (2G/3G/4G/5G), Bluetooth, NFC, GPS.
- **Touch-based input** – capacitive multi-touch screen, on-screen keyboard, voice input and gestures.
- **Battery power** – rechargeable Li-ion/Li-polymer battery; power efficiency is a key design goal.
- **Rich sensors** – accelerometer, gyroscope, magnetometer (compass), proximity, ambient light, barometer, fingerprint and face sensors.
- **Cameras and multimedia** – front/rear cameras, microphone, speaker, video recording.
- **Limited resources** – smaller screen, less RAM/CPU/storage than desktops, so apps must be lightweight.
- **Location awareness** – GPS, Wi-Fi and cell-tower positioning enable maps and location-based services.
- **App ecosystem** – applications are installed from stores such as Google Play and the Apple App Store.

Categories of Mobile Devices

Category	Description	Examples
Feature phone	Basic calls, SMS, limited apps; physical keypad	Nokia 3310, Nokia 105

Category	Description	Examples
Smartphone	Full OS, touchscreen, app stores, internet	iPhone, Samsung Galaxy, Xiaomi
Tablet	Larger screen (7–13 inch) for media and productivity	iPad, Galaxy Tab
Phablet	Between phone and tablet (5.5–7 inch)	Galaxy Note series
Wearable	Worn on the body; fitness and notifications	Apple Watch, Mi Band, Wear OS watches
E-reader	E-ink display optimised for reading	Amazon Kindle
PDA	Early handheld organiser (now obsolete)	Palm Pilot, HP iPAQ
Foldable	Flexible display that folds	Galaxy Z Fold, Z Flip

History of Mobile Devices

Year	Milestone
1973	Martin Cooper (Motorola) makes the first handheld mobile phone call.
1983	Motorola DynaTAC 8000X – first commercial handheld cellular phone (1G, analog).
1992	GSM (2G, digital) launches; first SMS text message sent. IBM Simon (1994) is regarded as the first smartphone.
2000s	3G networks enable mobile internet; BlackBerry and Nokia Symbian phones become popular.
2007	Apple releases the first iPhone with a multi-touch interface.
2008	First Android phone (HTC Dream / T-Mobile G1) released; Google Play (Android Market) opens.
2010s	4G/LTE, tablets (iPad 2010), smartwatches and app economy grow rapidly.
2019 onward	5G rollout, foldable phones and AI-powered features.

Note: Nokia's Symbian and Microsoft's Windows Phone were once major platforms but have been discontinued; today the market is dominated by Android and iOS.

1.2 Brands, Models and Platforms

Popular Brands and Models

Brand	Country	Sample Models	Platform
Apple	USA	iPhone 15/16, iPad	iOS / iPadOS

Brand	Country	Sample Models	Platform
Samsung	South Korea	Galaxy S series, A series, Z Fold	Android (One UI)
Xiaomi	China	Redmi, Poco, Xiaomi 14	Android (HyperOS/MIUI)
Google	USA	Pixel series	Android (stock)
Huawei	China	P series, Mate series	Android / HarmonyOS
OnePlus / Oppo / Vivo	China	OnePlus 12, Reno, V series	Android (OxygenOS/Color OS/Funtouch)
Nokia (HMD)	Finland	Nokia G/C series	Android

Mobile Platforms (Operating Systems)

A **mobile platform** is the operating system together with its development framework, tools and app distribution channel. The major platforms are:

Platform	Owner	Kernel / Language	Distribution
Android	Google (Open Handset Alliance)	Linux kernel; Kotlin / Java	Google Play, third-party stores
iOS	Apple	Darwin (XNU); Swift / Objective-C	Apple App Store only
HarmonyOS	Huawei	Microkernel; ArkTS	AppGallery
KaiOS	KaiOS Technologies	Linux/Firefox OS based; HTML, JS	KaiStore (feature phones)
Windows Phone (obsolete)	Microsoft	C#, .NET	Discontinued in 2019
BlackBerry OS (obsolete)	BlackBerry	Java, C++	Discontinued in 2022

Note: Android and iOS together hold more than 95% of the global smartphone market.

1.3 Introduction to Mobile Programming

Mobile programming (mobile application development) is the process of designing, coding, testing and publishing software applications that run on mobile devices. It differs from desktop development because of small screens, touch input, limited battery and memory, intermittent connectivity and the variety of device sizes.

Types of Mobile Applications

Type	Description	Pros / Cons
Native	Written for one platform with its official language and SDK (Kotlin/Java for Android, Swift for iOS)	Best performance and full hardware access; separate code per platform
Web app	Mobile-optimised website run in a browser (HTML, CSS, JavaScript)	No installation, one code base; limited device access, needs internet
Hybrid	Web code wrapped in a native container (Ionic, Cordova)	One code base; slower than native
Cross-platform	One code base compiled to native UI (Flutter, React Native, Xamarin/MAUI)	Fast development for both platforms; some platform-specific work still needed

Mobile App Development Process

- **Idea and requirement analysis** – identify the problem and target users.
- **Design** – wireframes, UI/UX, navigation flow.
- **Development** – coding front-end, back-end/API and database.
- **Testing** – unit tests, UI tests, device and performance testing.
- **Deployment** – publishing to Google Play / App Store.
- **Maintenance** – bug fixes, updates and user feedback.

Challenges in Mobile Programming

- Device fragmentation (many screen sizes, OS versions, manufacturers).
- Limited battery, memory and processing power.
- Unreliable or slow network connections.
- Security and privacy of user data; runtime permissions.
- Different platform guidelines (Material Design vs Human Interface Guidelines).

Example 1 — Choosing an approach

A college wants a student-notice app for both Android and iPhone users with a small team and short deadline. Which development type is suitable?

Solution

A **cross-platform** approach (e.g., Flutter or React Native) is suitable: one code base serves both platforms, reducing time and cost. A native approach would need two separate code bases and two developer skill sets.

Key Takeaways

- ✓ A mobile device is a portable, wireless, battery-powered computer; categories include smartphones, tablets, wearables and feature phones.
- ✓ Mobile history: 1G analog (1983) → 2G digital/SMS → 3G internet → 4G/LTE → 5G; iPhone (2007) and Android (2008) began the smartphone era.
- ✓ Android (Google) and iOS (Apple) are the dominant platforms.
- ✓ Mobile programming builds native, web, hybrid or cross-platform apps and must handle small screens, battery limits and connectivity.

— *End of Unit 1* —

bcalink.com