

DMAD-2-01P: Introduction to Windows Mobile and IOS Lab

Total Marks: 50

External Marks: 35

Internal Marks: 15

Credits: 2

Pass Percentage: 40%

P1: Hello, Windows! App

- **Tools:** UWP (Universal Windows Platform)
- **UI Element:** `TextBlock`

```
<TextBlock Text="Hello, Windows!" FontSize="24"
HorizontalAlignment="Center" VerticalAlignment="Center"/>
```

P2: Multi-Page Navigation

- **Use:** `Frame.Navigate()` with Button and Pages.
- **Controls:** `BottomNavBar`, `SplitView`, `Pivot`

P3: Local Storage (SQLite/File)

- **Option 1:** SQLite with `sqlite-net-pcl`
- **Option 2:** File I/O using `Windows.Storage.ApplicationData`

P4: Camera Capture

- **Tool:** `CameraCaptureUI`
- **Code Sample:**

```
CameraCaptureUI dialog = new CameraCaptureUI();
StorageFile photo = await
dialog.CaptureFileAsync(CameraCaptureUIMode.Photo);
```

P5: Map and Location

- **Tool:** `Geolocator` + `MapControl`

- **Permissions:** Location capabilities in manifest

iOS Apps (Swift/SwiftUI)

P6: Hello, iOS! App

- **Tool:** SwiftUI

```
Text("Hello, iOS!").font(.largeTitle).padding()
```

P7: Table View with Delegation

- **Tool:** UITableView, UITableViewDelegate, UITableViewDataSource
- **Use Case:** List of contacts/items

P8: Core Data

- **Features:** Add/Edit/Delete
- **Tool:** NSManagedObject, NSPersistentContainer

P9: Photo Library Picker

- **API:** UIImagePickerController
- **Permissions:** Add NSPhotoLibraryUsageDescription in Info.plist

P10: MapKit Integration

- **Tool:** MKMapView
- **Function:** Show pins, track user, directions

Cross-Platform (Flutter/Xamarin/React Native)

P11: Responsive App (Flutter/Xamarin.Forms)

- **Layout:** Use MediaQuery, LayoutBuilder, or Device.Idiom
- **UI Frameworks:** Flutter or MAUI (Xamarin successor)

P12: Push Notifications

- **Tools:**
 - iOS: APNs + Firebase Cloud Messaging
 - Windows: WNS + Firebase
 - Cross-platform: Firebase with abstraction layer

P13: Data Synchronization

- **Solution:** Firebase Realtime DB or Cloud Firestore
- **Sync:** Realtime listeners + shared user token

P14: Offline Functionality

- **Techniques:** SQLite, cached storage, service workers
- **Frameworks:** WorkManager (Flutter), Reachability (iOS)

P15: In-App Purchases

- **Platforms:**
 - iOS: StoreKit
 - Windows: StoreContext
 - Cross-platform: `in_app_purchase` (Flutter)

P16: AR Features

- **Windows:** Mixed Reality Toolkit (MRTK)
- **iOS:** ARKit (e.g., plane detection, image tracking)
- **Cross-platform:** Unity or Vuforia

P17: Biometric Authentication

- **iOS:** `LAContext` (Face ID/Touch ID)
- **Windows:** Windows Hello API
- **Cross-platform:** `local_auth` plugin in Flutter

P18: Accessibility Enhancements

- **iOS:** VoiceOver support with `accessibilityLabel`
- **Windows:** Narrator support using `AutomationProperties.Name`

P19: Custom Animations

- **iOS:** `UIView.animate` or SwiftUI's `.animation`
- **Windows:** `Storyboard`, `VisualStateManager`
- **Flutter:** `AnimatedContainer`, `Hero`, `TweenAnimationBuilder`

P20: Cloud Integration

- **Cloud Providers:** Firebase (Firestore, Auth), Azure (App Services, Cosmos DB)
- **Auth + Sync:** Use SDKs with real-time listeners and cloud functions