CSIT242 Mobile Application Development

LECTURE 7-2 – REVISION

Assessment

Assessment Items	Percentage of Final Mark		Due Date
	Marks for the Item	Minimum required for a pass	
Lab Exercises 1-5	6 each	N/A	As scheduled
Assignment 1 & 2	10+10 = 20		As scheduled
Final Exam	50	20	Exam week as per schedule
Total	**100**	**50**	**The mark must be ≥50 to pass the subject**

Technical Fail

- To be eligible for a Pass in this subject a student must achieve a mark of at least 40% in the
  - Final Exam (20 out of 50)

- Students who fail to achieve this minimum mark & would have otherwise passed may be given a TF (Technical Fail) for this subject

Supplementary Exams

- The School does not normally offer a supplementary exam to a student who has sat a scheduled exam.
- Supplementary Exams will be dealt with in accordance with student academic consideration policy
- While the School normally grants supplementary exams when the student does not sit the standard exam for an acceptable reason, each case will be assessed on its own merit and there is no guarantee a supplementary exam will be granted.
- If a supplementary exam is granted, you will normally be notified with the time and date of this supplementary exam. You must follow the instructions given in the email and messages.

Final Exam

Duration: 3 Hours

The exam is closed book - You are NOT allowed to use any materials during the exam

Academic Integrity is important!

Final Exam

Part A: 10 fill-in questions

- Fill in the blanks to create a correct statement
- Answer all questions for both Android and iOS
- 10 x 1.5 mark each = 15 marks

Part B: 5 theory questions

- Answer all questions and sub-questions for both Android and iOS
- 5 x 3 marks each = 15 marks
  - What is …
  - What are …
  - Explain …
  - Compare and contrast ….
  - What are similarities / differences ….

Final Exam

Part C: 5 coding questions

- you can choose Android or iOS
- **2 questions** explain the code segment (2 x 2.5 marks each = 5 marks)
  - answer in detail. What is the purpose and the output produced by the following code fragment? Explain your
- **3 questions** write code segment (3 x 5 marks each = 15 marks)
  - Write a code fragment that will solve the problem given below. Write additional comments to your code and explain your answer in detail.

    Select only one of the following problems to write the code (Android or iOS).

Final Exam

Part A: 10 fill-in questions

A TabBar in iOS appears at the [ ] of an app screen and provides the ability to quickly [ ] between different sections of an app (UITabBar). Tab bars are [ ], may have a background tint, maintain the same height in all screen orientations, and are [ ] when a keyboard is displayed. A Tab bar may contain any number of tabs, but the number of visible tabs varies based on the device size and orientation. If some tabs can't be displayed due to limited horizontal space, the final [ ] becomes a More tab, which reveals the additional tabs in a list on a [ ] screen.

bottom | switch | translucent | hidden | visible tab | separate screen | top screen | toolbar | top

Part B: 5 theory questions

What are the similarities and differences between MVC and MVP design patterns?

Final Exam

Part C: 2 questions explain the code segment

What is the purpose and the output produced by the following code fragment? Explain your answer in detail.

a) Android (2.5 marks)

```
//public class databaseHelper
. . .
public Integer deleteRecord(String id){
   SQLiteDatabase db = this.getWritableDatabase();
   return  db.delete("MyTable", "ID = ?",new String[] {id});
}
. . .

// MainActivity.java - method myMethod()
databaseHelper myDB = new databaseHelper(this, null, null, 1);
EditText txtID = (EditText) findViewById(R.id.txtID);
Integer nRow = myDB.deleteRecord(txtID.getText().toString());
if (nRow > 0) {
    Toast.makeText(MainActivity.this, "Success!", Toast.LENGTH_LONG).show();
} else {
    Toast.makeText(MainActivity.this, "Error", Toast.LENGTH_LONG).show();
}
```

b) iOS (2.5 marks)

```
//file MyViewController.swift
class MyViewController: UIViewController, MKMapViewDelegate {
@IBOutlet weak var mapView: MKMapView!
. . .
let userLocation = mapView.userLocation
let region = MKCoordinateRegion(center: userLocation.location!.coordinate,
                                latitudinalMeters: 1000, longitudinalMeters: 1000)
mapView.setRegion(region, animated: true)
. . .
}
```

Final Exam

Part C: 3 coding questions write code segment

Write a code fragment that will solve the problem given below. Write additional comments to your code and explain your answer in detail.

Select only one of the following problems to write the code (Android or iOS) (5 marks)

a) Android - Write a code fragment that will present a Toast notification to the user (for a more extended period of time) with the message "This is a good option".

b) iOS - Write a code segment to present an Action Sheet to the user. The title should be "My action sheet" with the message "Is this a good option?" and three buttons "Yes", "No" and "Try Again".

Subject Outline

- Gain knowledge for mobile application
  - design,
  - development,
  - implementation and
  - deployment.
- Examine different mobile platforms and learn how to use.
- Develop technical skills necessary to develop applications using
  - different languages,
  - frameworks and
  - tools.

Lecture 1

- Introduction to Android and iOS
  - OS Architecture (layers)
  - Application behaviour
  - Development tools

Lecture 1

- Design patterns
  - MVC
  - MVP
  - MVVM
  - Delegation

- Application architecture
  - App life cycle / app's states
  - Android manifest / info.plist
  - Components (activities, services, broadcast receivers, intents,…) / Objects (UIApp object, App Delegate, Data Model, …)
  - Requirements; Resources

Lecture 2

- User interface
  - Layouts / Scenes (StoryBoard)
  - Controls / Views
  - Events, Event Listeners and Callback methods, Event Handlers
  - Outlets and Actions

Lecture 3

- Java syntax
  - Dalvik VM
  - Android Run Time
- Swift syntax
  - variables, constants, arrays, tuples
  - optionals
  - statements
  - functions, closure
  - classes, objects

Lecture 3

- Android
  - styles
  - themes
  - menus, dialogs, fragments
  - notifications, alerts, toasts, snackbars, ….
- iOS
  - toolbars / tab bars
  - action sheets, alerts
  - notifications, popovers

Lecture 4

- Storage options in Android
  - App-specific files
  - Shared storage (Media & Documents and other files)
  - Shared preferences
  - Databases (SQLite, Room)
  - Content providers

- iOS persistent storage
  - working with files (Application Support Directory, Documents Directory, Temp Directory)
  - iCloud storage
  - iOS Core Data framework
    - Core Data Stack

Lecture 5

- Android
  - view animation,
  - property animation,
  - transitions
- Android Audio and video
  - Multimedia framework
  - MediaPlayer
  - Camera
- iOS
  - animations,
  - transitions,
  - Core Animation
- iOS Audio and video
  - Media layer
    - Core Graphics
    - Core Image
    - AVFoundation
  - Camera

Lecture 6

- Location and maps
  - Geocoding
  - Android location, Google Map API
  - Core Location, Map Kit
- Sensors
  - Categories / types of sensors
  - Core Motion

Lecture 6

- iOS Frameworks
  - GameKit, GameplayKit
  - HealthKit
  - Core Bluetooth
- Android connectivity
  - Bluetooth
  - NFC
  - Wi-Fi P2P
  - USB
- Networking
- App Testing
- App publishing

Lecture 7

- Multithreaded applications
- App's performance and optimization
- Security

Notes

- Review all lectures and labs
- Review the lab exercise solutions