MODULE NAME:	MODULE CODE:
OPEN SOURCE CODING INTRODUCTION	OPSC7311

ASSESSMENT TYPE: ASSIGNMENT 2 (PAPER ONLY)

TOTAL MARK ALLOCATION: 100 MARKS

TOTAL HOURS: 15 HOURS

STUDENT NAME:

STUDENT NUMBER:

INSTRUCTIONS:

1. ***Any assignment with a similarity index of more than 25% will be scrutinized for plagiarism.***
2. ***Make a copy of your assignment before handing it in.***
3. *Assignments must be created in Android Studio and source code must be stored electronically on a CD.*
4. *All work must be adequately and correctly referenced.*
5. *Follow all instructions on the assignment cover sheet.*
6. *This is an individual assignment.*
7. ***This is a non-specific assignment; therefore, marks will be awarded for creativity and effort.***
8. ***Please ensure that you keep backups of all source code and documentation submitted. In your Portfolio of Evidence you will be required to make changes to your submission based upon feedback given by your lecturer.***

Referencing Rubric

Providing evidence based on valid and referenced academic sources is a fundamental educational principle and the cornerstone of high quality academic work. Hence, The IIE considers it essential to develop the referencing skills of our students in our commitment to achieve high academic standards.

Poor referencing will result in a penalty of a maximum of 5 marks against the percentage mark awarded to a student according to the following guidelines. Please note that this will not hold for assignments in which referencing is not required.

Required	Subtract 1	Subtract 2	Subtract 3	Subtract 4	Subtract 5
• Consistent, in-text referencing style • Quotation marks, page numbers, years, etc. applied correctly • Only one or two minor mistakes made • All sources academically sound and included in reference list • Accurate recording of references.	• Consistent in-text referencing style • Quotation marks, page numbers, years, etc. applied correctly • Fewer than five minor mistakes made • More than 90% of the sources are academically sound and included in reference list • Mostly accurate recording of references.	• Consistent in-text referencing style • Quotation marks, page numbers, years, etc. not always applied correctly • Not all paraphrased content referenced • At least 80% of the sources are academically sound and included in reference list • Generally accurate recording of references.	• Consistent in-text referencing style • Quotation marks used for direct quotes but page numbers missing • Fewer than 50% of the sources are academically valid • Reference list incomplete • Inaccurate recording of references.	• Referencing style inconsistent • Paraphrased material not referenced in text • Quotation marks and/ or page numbers for direct quotes missing • Fewer than 25% of the sources are academically valid • Reference list incomplete.	• Exceptionally poor and inconsistent referencing • None of the sources academically valid.

The section below applies to both Question 1 and 2.

General Requirements

Writing maintainable code in industry is vital. Therefore, you are required to:

1. Follow good programming practices by ensuring that the following are correct:
   - variable types;
   - variable scope; and
   - Class and Method naming standards.
2. Insert comments to show logic and design for the support of code maintainability.
3. Code efficiently. Redundant code must be avoided.
4. Document your application thoroughly. Although you are not required to submit user and developer documentation with this assignment, you are required to submit these with your Portfolio of Evidence. Therefore it makes good sense to create this documentation while you are in the process of developing these apps.
5. The types of documentation you should at least consider are:
   a. Description of the app. (Describe the problem and how it can be solved using a file / database based system.
   b. Program flowcharts or alternatively pseudo-code.
   c. UML Diagrams (at least class and use case diagrams)
   d. Test Data used to debug the application
   e. Help files/ user guides for the system.
   f. Screenshots of the output produced by the app.

Requirement	**Maximum Mark**	**Marker**	**Moderator**
Good programming practice (Naming standards, indentation, variable types and scope.) • Not followed at all = 0 • Some standard adhered to (at least 2) = 1 • Most standards adhered to (at least 3) = 2 • Excellent – all standards adhered to = 3	(3)		
Requirement	**Maximum Mark**	**Marker**	**Moderator**
Code efficiency	(2)		

• Poor –multiple instances where code is duplicated = 0 • Average – some redundant code exists = 1 • Excellent – code very efficient = 2			
Comment statements • Poor – No comments = 0 • Few comments (only describing methods/variables) = 1 • Some comments (only describing logic) = 2 • Excellent – all necessary comments given to explain variables, methods and logic = 3	(3)		
Program compiles and executes • No – Many changes required = 0 • Average – Minor changes required = 1 • Yes – No changes required = 2	(2)		
Total Marks	**10**		

[Mark Allocation: 10]

Before you attempt this assignment ensure that you have Android Studio installed and working. You would also have completed various activities and viewed the video tutorials. Using Android Studio create a new Android Project named OPSC7311_Assign2_StudentNumber

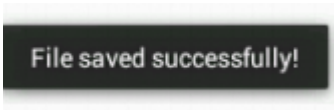
Please note that there is no lecture contact time for the completion of this assignment. You need extensive out-of-class hours to complete this assignment.

Question 1 (Marks: 40)

Within this Android application name the first activity Question One and create code and the GUI to perform the following:

Q.1.1 Choose any problem that can be solved using a file-based application. Examples can include but not limited to a reservation system, a ticketing system, an address book, favourite places, tourist attractions in a specific city, top ten lists, events management, etc. You will need to conceptualise, design and code this app as follows:

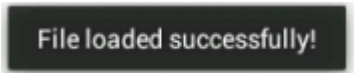
- Design the files required for your system. You will need to store data and produce at least one report. The minimum requirement is one file and one report, but you may use as many as you deem necessary.
- Your file must contain records.
- Design a GUI interface in XML using "View" components to capture new records as well as to display your report(s).
- Write code to allow a user to save all details from the GUI into a text file. Print out a success message detailing that the information has been successfully saved. Use a message (Toast class/ SnackBar or Notifications Bar) to indicate the success or failure of the file save as follows:



- You may want to view your textfile and verify that it was created and saved. A search through your computer or within the project source files will not show this file. You need to use the DDMS. Use your text resources to assist you in finding and "pulling" this file to view it.

- You also need functionality to open the text file and display the information in a new Activity or Intent. This could be your report.

- Also display a message to confirm that all data has been loaded. You can use the Toast class or ToolBar or SnackBar.



- Use extensive exception handling to determine if the files (or any input) are invalid. Make use of the Toast class or ToolBar or AppBar to display error messages.

- Research why your app does not need a traditional **Exit** button. However there is still a need to clear the stack of activities through programming and also stop any services and processes, e.g. GPS recording, which may be running in the background. Include this functionality in your app.

- Your app needs to include proper navigation as specified by the Google guidelines and standards.

- Feel free to add extra functionality into the program. You will be awarded appropriately.

- The above specifications are just a guideline; you may adapt this to your own styling and add other views like images, etc. You are also encouraged to increase the functionality; however the above is the minimum specification.

N.B: 1. You are not allowed to combine this question with the next one.

2. Your idea/ design must be approved by your lecturer before you start coding.

Save your activity files as **Question1.**

Question 2 (Marks: 50)

Within your Android Assignment 2 project create a second new Activity named Question2.

Using Question 1 as a guideline and example, you are now required to create a database application. You need to conceptualise, design and code your own database and app based on your own idea. Try to make your database app useful to a specific business, non-profit organisation (NGO) or education. Plagiarism by downloading designs and code from the Internet and adding it to your assignment is not permitted. You will be penalised for this and your assignment will not be marked.

You are free to choose any application area, which requires the storage of information in a database. You are required to build and create the GUI XML file and the Java code in Android Studio.

Some examples that may be used as Database applications include top ten lists, storing module assessment marks, storing information of business contacts, store user preference data, store the co-ordinates of the location that a user has been to; storing club names, their locations and ratings, customer databases, product catalogues, sporting teams, scientific tables etc. You may also use the concept you created in question 1 as the same idea for this question.

N.B: 1. You are not allowed to combine the two questions into one.

2. Your idea/ design must be approved by your lecturer before you start coding.

Your Android application must satisfy the following specifications:

- Create the "DBAdapter" helper class as specified in the course material.
- Your database can be created programmatically or pre-created, using SQLite.
- Include an activity that specifies "How to Use" your application.
- Your database app must include the four (4) CRUD functions i.e. add a record, update, delete and modify/ change a record. You may use menus or buttons to implement these functions.
- You may also include search/ retrieval functionality and reports. However this is optional.
- Methods and parameter passing must be used.
- A GUI designed using the XML file (you are free to use any views e.g. "EditText", radio buttons, checkboxes, combo boxes, etc. use your creativity in the design).
- Validate all input.
- Keep your application simple and restricted to one area.

[Total Marks: 100]