Module	**ITC 3234 OBJECT ORIENTED PROGRAMMING (Level 5)**		
Term	**Spring Semester 2020**		
Assessment	**PROGRAMMING PROJECT**	**Weight**	**60%**
Relevant Learning Outcomes	3. Model software requirements using UML. 4. Design, implement, and test advanced, distributable, and maintainable object-oriented GUI applications.		
Deliverables	Part 1: Proposal & Draft UMLs Part 2: Source Code Part 3: Report		
Method of Submission	Online through Blackboard		
Professor	Maira Kotsovoulou		
Deadlines:	***Part 1 – Draft Project Proposal and UML: March 12th, 2020 (14:00)*** *(If part 1 is not submitted on time, will be penalized by 10% for each day of delay - max delay 6 working days. Students will lose the percentage of part 1 if work is not submitted within the deadline)* ***Parts 2 + 3: April 2nd, 2020 (23:59)*** *Work submitted late and within 6 working days after the deadline, will be penalized by 10% for each day of delay to a minimum of **40** points (for passing grades). Students will fail the assessment if work is submitted after the late submission deadline.*		

Please note that College regulations regarding academic integrity apply.

Design and implement a GUI application of a topic of your choice
The application may cover all or part of a real-life application needs but should be able to work in a stand-alone fashion.

PART 1: PROJECT PROPOSAL & UMLS (LO 3: 6 POINTS)

Step 1
Prepare and submit to your instructor a short proposal, which must be **approved before proceeding**. The proposal must include the following:

- a description of the topic of the application (refer to the context and the objective).
- draft requirements, in the form of a behavioural (use-case) UML diagram.
- draft design, in the form of a class UML diagram of the classes you are considering, and which will reflect the requirements.

Note: In case your proposal is not approved, you will be requested to submit a new one.

PART 2: APPLICATION (LO 4: 74 POINTS)

Create a java application, which fulfills the following requirements:

Functionality requirements (70 points)

Your application should utilize a graphical user interface and include:

1. Inheritance (1 parent and two children object classes);
2. 3 Object classes with:
   a. appropriate constructors
   b. setters and getters
   c. method/s to perform some other functionality/calculation (for example calculate VAT or salary TAX or product Discount price… etc.)
   d. toString()
3. An enumerator;
4. Login form (utilizing user authorization and authentication processes);
5. Menu for navigation;
6. Data manipulation mechanisms:
   a. add
   b. modify
   c. delete
7. Data validation during add and update using appropriate error messages;
8. Data presentation using JTables;
9. Search functionality
   a. single – exact match – criterion
   b. single criterion – with “includes”
   c. multiple criteria using and / or
10. Basic file manipulation (read data and save data to file/s)
    a. File Chooser for file open and file save
    b. Automatically when the application starts using default location and file name
11. Display statistics for one piece of data (meaningful with regard to context);

12. One extra context-related functionality of your choice;
13. Javadoc for the object classes

Implementation requirements (Which will subtract point)

Your code must:

1. compile successfully (-10points if it does not);
2. exit properly (should not crash) (- 10 points if it crashes)

Packaging requirements (5 points)

Create a jar file with all your code, which will be the executable version of your application.

PART 3: DOCUMENTATION & REPORT (20 POINTS)

Prepare and submit documentation regarding your application, which should include the following:

1. General information about your topic and your code design
   a. A description of your topic, integrating the feedback you have received for the respective component of your project proposal (part 1).
   b. A list of the classes you have designed with a short description for each one of them, stating their purpose.
2. Modelling information:
   a. Use case UML diagram (using actors) that accurately models the application you have eventually developed, accompanied by a comparison with the pre-development version (part 1).
   b. A java class UML diagram, which accurately reflects the dynamics between the classes of the application you have developed, accompanied by a comparison with the pre-development version (part 1).
   c. A list of all the java classes that you have used (and where they are being used).
3. Screenshot of a sample execution of the application
4. Reflections and evaluation about:
   a. features you have successfully or partially implemented;
   b. difficulties you faced during development;
   c. features you have not yet implemented but you wish to implement in future versions;
   d. how the modelling process has assisted or hindered the development of your application.
5. Testing information
   a. Sample data/runs and results. If you have used sample files, you should submit them online through BB.

Use a table of contents with appropriate headings.
Include references, in case you utilized or adapted code or libraries from someone else.

MARKING SCHEME

Task		Criteria
Project Proposal		**6**
Topic description	2	none, limited, satisfactory, good, excellent
UML Modelling: Use Case Diagram	2	none, limited, satisfactory, good, excellent
UML Modelling: Class Diagram	2	none, limited, satisfactory, good, excellent
APPLICATION DEVELOPMENT		**74**
Object Oriented Concepts		
Inheritance	4	1 parent and two children object classes
3 Complete object classes	6	constructors, setters and getters, utility method
Javadoc for object classes	3	none, some, always
Enumerator	3	creation and appropriate use in object class and in GUI
GUI Elements		
Login Form	4	none, simple, with logo, with privileges
Menu	4	none, non-functional, functional, with about dialog
JTables	5	none, populated with data from collections, can sort data
Data Manipulation & Validation		
ADD	5	none, designed but does not work, functional
DELETE	5	none, designed but does not work, functional
UPDATE	5	none, designed but does not work, functional
Data Validation	5	none, limited, some validation, full validation
Search		
Exact match	3	none, some code exists but does not work, works well
Includes	3	none, some code exists but does not work, works well
Multiple fields (AND/OR)	2	none, some code exists but does not work, works well
File Manipulation		
Basic	4	Automatic in application open and close (default filename)
Utilizing a File Chooser	3	none, file chooser is there but does not work, file open works, file save works, both work
Other Application components		
Display statistics	4	none, limited, satisfactory, good, excellent
Extra Functionality	4	none, limited, satisfactory, good, excellent
Packaging (deploy to jar)	5	yes, no
Documentation & Report		**20**
Topic description	4	none, limited, satisfactory, good, excellent
UML Modelling: Use Case Diagram	4	none, limited, satisfactory, good, excellent
UML Modelling: Class Diagram	4	none, limited, satisfactory, good, excellent
Reflections	4	none, limited, satisfactory, good, excellent mentioning future improvements
Testing information Screen shots sample code execution	4	none, limited, satisfactory, good, excellent

Please note that College regulations regarding academic integrity apply.