FACULTY OF INFORMATION TECHNOLOGY

PROGRAMMING 512

SEMESTER 2 ASSIGNMENT

Name & Surname: _____ ITS No: _____

Qualification: _____ Semester: _____ Module Name: _____

Date Submitted: _____

ASSESSMENT CRITERIA	MARK ALLOCATION	EXAMINER MARKS	MODERATOR MARKS
MARKS FOR CONTENT			
QUESTION ONE	25		
QUESTION TWO	25		
QUESTION THREE	40		
TOTAL	90		
MARKS FOR TECHNICAL ASPECTS			
TABLE OF CONTENTS	2		
ASSIGNMENT LAYOUT	5		
REFERENCES	3		
TOTAL MARKS FOR ASSIGNMENT	100		
EXAMINERS COMMENTS:			
MODERATORS COMMENTS:			
SIGNATURE OF EXAMINER:		SIGNATURE OF MODERATOR:	

ASSIGNMENT INSTRUCTIONS

1. All assignments must be typed, not handwritten.
2. Every assignment should include the cover page, table of contents and a reference list or bibliography at the end of the document.
3. A minimum of three current sources (references) should be used in all assignments, and these should be reflected in both in-text citations and the reference list or bibliography.
4. In-text citations and a reference list or bibliography must be provided. Use the Harvard Style for in-text citations and the reference list or bibliography.
5. Assignments submitted without citations and accompanying reference lists will be penalised.
6. Students are not allowed to share assignments with fellow students. Any shared assignments will attract stiff penalties.
7. Using and copying content from websites such as chegg.com, studocu.com, transtutors.com, sparknotes.com or any other assignment-assistance websites is strictly prohibited. This also applies to Wiki sites, blogs, and YouTube.
8. Any pictures and diagrams used in the Assignment should be labelled appropriately and referenced.
9. Correct formatting of answers (font size 12, font style Calibri, line spacing of 1.15 and margins justified). Each question should begin on a new page.
10. All assignments must be saved in PDF using the correct naming convention before uploading them to Moodle: E.g., StudentNumber_CourseCode_Assignment (402999999_PRO512A_Assignment).

Question 1 (25 marks)

1. Explain how the following set operations work: (7 marks)
   a. Union
   b. Intersection
   c. Difference
2. Discuss the four characteristics of Object-Oriented Programming. (8 marks)
3. Define the following terms: (10 marks)
   a. Database
   b. Record
   c. Field
   d. Primary Key
   e. Foreign Key

Question 2 (25 marks)

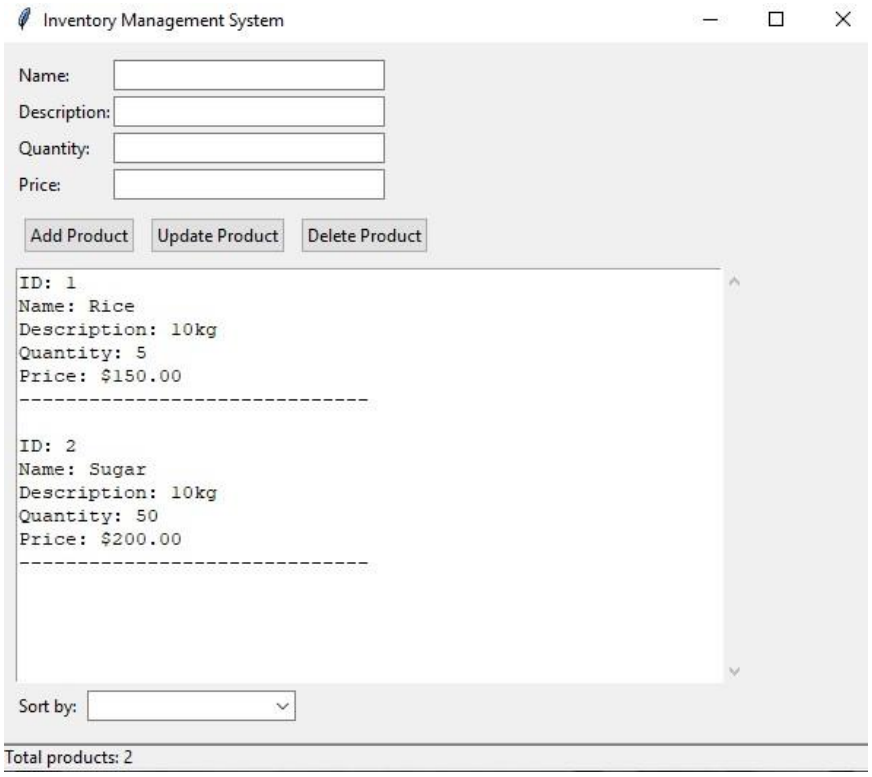
Implement a University Course Management System using Object-Oriented Programming principles in Python. The system should manage courses, students, and professors. Create the following classes with their respective attributes and methods:

a. Create a ***Person*** class with name, age, and ID attributes. Implement an init method to initialise these attributes and a str method to return a string representation of a Person object.
b. Develop a ***Student*** class that inherits from the Person class. In addition to the inherited attributes, include a major and a list of enrolled_courses. Implement an enrol(course) method that adds a course to the student's enrolled_courses if there's capacity in the course. Also, create a drop(course) method that removes a course from the student's enrolled_courses.
c. Create a ***Professor*** class that inherits from the Person class. Add attributes for the department and a list of courses_teaching. Implement an assign_course(course) method that adds a course to the professor's courses_teaching list and sets the course's professor attribute to this professor.
d. Design a ***Course*** class with attributes for course_code, name, max_capacity, professor, and a list of enrolled_students. Implement init and str methods, add_student(student) and remove_student(student) methods. The add_student method should add a student to the course if there's capacity, while the remove_student method should remove a student from the course.
e. Develop a ***University*** class with attributes for names and lists of courses, students, and professors. Implement methods to add_course(course), add_student(student), and add_professor(professor) to their respective lists. Also, create a get_course(course_code) method that returns a course given its course code.

Create a demo script that instantiates the University class and populates it with courses, students, and professors. Demonstrate the functionality by enrolling students in courses, assigning professors to courses, dropping students from courses, and printing out the state of courses, students, and professors after these operations.

Question 3 (40 marks)

Develop a Python application for managing product inventory using Tkinter for the graphical user interface and SQLite for database storage.



Requirements:

1. Database Setup:
   - Use SQLite to create a database named "inventory.db"
   - Create a table named "products" with the following columns:
     - id (Integer - Primary Key)
     - name (Text)
     - description (Text)
     - quantity (Integer)
     - price (Real)

2. Graphical User Interface:
   - Create a main window with the title "Inventory Management System"
   - Implement the following Tkinter widgets:
     - Entry widgets for product details (name, description, quantity, price)
     - Buttons for Add, Update, and Delete operations
     - A Text widget to display all product details
     - A Scrollbar for the Text widget
     - A Combobox for sorting options
     - A Label to show the total number of products (status bar)
3. Functionality:
   - Implement functions to perform the following operations:
     - Add a new product to the database.
     - Update an existing product's details.
     - Delete a product from the database.
     - View all products in the Text widget.
     - Sort products by name, price, or quantity (ascending and descending)
   - Ensure proper input validation for all entry fields.
   - Implement error handling with appropriate message boxes.