DEPARTMENT OF MECHANICAL AND INDUSTRIAL ENGINEERING
MECHANICAL ENGINEERING PROGRAM, BSC.

Course Syllabus

1. **Course number and name**
   IE 803233 Object Oriented Programming for Industrial Engineering

2. **Credits and contact hours**
   (3+0) 3 credit hours, 3 contact hours

3. **Course type**
   Blended Learning Course (2+1)

4. **Instructor's or course coordinator's name**
   Eng. Dana Salameh

5. **Textbook information**
   Fundamentals of Python Programming. Richard L. Halterman. 2018

   a. **Other supplemental materials**
   Foundations for Analytics with Python. Clinton W. Brownley. 2016

   Kenneth A. Lambert, The Fundamentals of Python: First Programs, 2011, Cengage Learning, ISBN: 978-1111822705.

6. **Specific course information**
   a. **Catalog description**
   General Introduction, Comments, Variables, Statements, and Expressions, Debugging, Python Modules, Python Turtle, Sequences, Iteration, Conditionals, transforming sequences, files, dictionaries, functions, classes and objects, building programs, inheritance

   Prerequisites or co-requisites
   Prerequisite: 0810201 Computer Skills 2

   b. **The course is:**
   Required in Industrial engineering program.

7. **Specific goals for the course**
   a. **Course outcomes:**

DEPARTMENT OF MECHANICAL AND INDUSTRIAL ENGINEERING
MECHANICAL ENGINEERING PROGRAM, BSC.

After completion of the course, students are expected to be able to understand:

1. Introduction to Python, use IDLE to develop programs, Basic coding skills, working with data types and variables, working with numeric data, working with string data, Python functions, Boolean expressions, selection structure, iteration structure, Illustrative Programs, Exercises
2. Define and use functions and modules, Basic skills for working with lists, tuples, sets, ranges, dictionaries
3. Object Oriented Programming, an introduction to classes and objects, define a class, work with object composition, work with inheritance
4. Python Turtle

    b. The following student outcomes are addressed by the course:
    None

8. **Learning Outcomes and their Alignment with Program Educational Objective (PEO's), Methods of Delivery, and Assessment Methods:**

Learning Outcomes	Program PEOs	Method of Delivery	Assessment Method
Course Outcomes			
CO-(1): Introduction to Python, use IDLE to develop programs, Basic coding skills, working with data types and variables, working with numeric data, working with string data, Python functions, Boolean expressions, selection structure, iteration structure, Illustrative Programs, Exercises	-	synchronous active learning	Midterm
CO-(2): Define and use functions and modules, Basic skills for working with lists, tuples, sets, ranges, dictionaries	-	Lectures (Example and Problems)	Final exam

DEPARTMENT OF MECHANICAL AND INDUSTRIAL ENGINEERING
MECHANICAL ENGINEERING PROGRAM, BSC.

CO-(3): Object Oriented Programming, an introduction to classes and objects, define a class, work with object composition, work with inheritance	-	Lectures (Example and Problems)	Project
CO-(4): Python Turtle	-	asynchronous active learning	Project
Student Outcomes			

9. Weekly Teaching Plan

Week No.	Lecture	Topic	Method of Delivery
1	Sun (9-10)	Python home Python Introduction	Lecture
	Tue (9-10)	Python Getting Started Python Syntax Python Indentation	Lecture
	Thu (9-10)	Python Comments Python Variables	synchronous active learning
2	Sun (9-10)	Python - Variable Names Python Variables - Assign Multiple Values	Lecture
	Tue (9-10)	Output Variables Python - Global Variables	Lecture
	Thu (9-10)	Variable exercises Python Data Types	synchronous active learning
3	Sun (9-10)	Python Numbers Order of Operations Python Casting	Lecture
	Tue (9-10)	Python Strings Python - Slicing Strings Python - Modify Strings	Lecture
	Thu (9-10)	Python - String Concatenation Python - Format - Strings	Online Lecture/ synchronous

4	Sun (9-10)	Python - Escape Characters	Lecture
	Tue (9-10)	Python - String Methods	Lecture
	Thu (9-10)	String exercises	Online Lecture
5	Sun (9-10)	Python Booleans Python Operators	Lecture
	Tue (9-10)	Python Lists Python - Access List Items Python - Change List Items	Lecture
	Thu (9-10)	Python - Add List Items Python - Remove List Items Python - Loop Lists	Online Lecture
6	Sun (9-10)	Python - List Comprehension Python - Sort Lists Python - Copy Lists	Lecture
	Tue (9-10)	Python - Join Lists Python - List Methods	Lecture
	Thu (9-10)	Python List Exercises Python Tuples	Online Lecture
7	Sun (9-10)	Python - Access Tuple Items Python - Update Tuples Python - Unpack Tuples	Lecture
	Tue (9-10)	Python - Loop Tuples Python - Join Tuples	Lecture
	Thu (9-10)	Python - Tuple Methods Python - Tuple Exercises	Online Lecture
8	Sun (9-10)	Python Sets Python - Access Set Items Python - Add Set Items	Lecture
	Tue (9-10)	Python - Remove Set Items Python - Loop Sets Python - Join Sets	Lecture
	Thu (9-10)	Python - Set Methods Python - Set Exercises	Online Lecture
9	Sun (9-10)	Python Dictionaries Python - Access Dictionary Items Python - Change Dictionary Items	Lecture
	Tue	Python - Add Dictionary Items	Lecture

	(9-10)	Python - Remove Dictionary Items Python - Loop Dictionaries Python - Copy Dictionaries	
	Thu (9-10)	Python Dictionary Methods Python Dictionary Exercises	Online Lecture
10	Sun (9-10)	Python If ... Else Python While Loops	Lecture
	Tue (9-10)	Python For Loops Python Functions	Lecture
	Thu (9-10)	Python Lambda Python Arrays	Online Lecture
11	Sun (9-10)	Python Scope Python Modules	Lecture
	Tue (9-10)	Python Datetime Python Math	Lecture
	Thu (9-10)	Python Try Except Python User Input	synchronous active learning
12	Sun (9-10)	Python String Formatting Python Random Module	Lecture
	Tue (9-10)	Python statistics Module	Lecture
	Thu (9-10)	Python math Module Python cmath Module	Asynchronous active learning
13	Sun (9-10)	File Handling Python File Handling Python Read Files	Lecture
	Tue (9-10)	Python Write/Create Files Python Delete Files	Lecture
	Thu (9-10)	Python Built in Functions Python Keywords	Asynchronous active learning
14	Sun (9-10)	Python Classes and Objects Python Inheritance	Lecture
	Tue (9-10)	Python Classes and Objects Python Inheritance	Lecture
	Thu (9-10)	Python Classes and Objects Python Inheritance` project	Asynchronous active learning

15	Sun (9-10)	Turtle Graphics	Lecture
	Tue (9-10)	Turtle Graphics	Lecture
	Thu (9-10)	Turtle Graphics project	Asynchronous active learning

Grade Distribution:

Assessment	Grade	Date
- First Exam	20%	Fifth Week
- Second Exam	20%	10th Week
- Assignments and Projects	10%	
- Final Examination	50%	16th Week

* Make-up exams will be offered for valid reasons. It may be different from regular exams in content and format