Department of Mathematics

College of Science

Salahaddin University-Erbil

Subject: Advanced Computer Skills

Course Book: PhD Degree

Lecturer's name: Imad A. Aziz

Academic Year: 2023-2024

Course Book

1. Course name	Computer Programming Python
2. Lecturer in charge	Imad A. Aziz
3. Department/ College	Mathematics / Science
4. Contact	e-mail: imad.aziz@su.edu.krd Tel: +9647504639909
5. Time (in hours) per week	For example Theory: 3 Practical: 0
6. Office hours	Sat. 11:30 – 2:30
7. Course code	
8. Teacher's academic profile	**23/6/2020** lecturer at Department of Mathematics, College of Science, University of Salahaddin-Erbil, Iraq. **16/6/2020** Awarded Ph.D. in Mathematics, Department of Mathematics, College of Science, University of Salahaddin-Erbil, Iraq. **3/9/2006** Assistant lecturer at Department of Mathematics, College of Science, University of Salahaddin-Erbil, Iraq. **31/7/2006** Awarded M.Sc. in Mathematics, Department of Mathematics, College of Science, University of Al-Mustansiriyah, Iraq. **10/1/2002** Awarded B.Sc. in Mathematics, Department of Mathematics, College of Science, University of Salahaddin-Erbil, Iraq. **1995-1996** Awarded Baccalaureate, Hamren Secondary Puleimaniyah (Kalar), Iraq.
9. Keywords	PyCharm, Tuples, Modules, Packages, NumPy, PyPlot

10. Course overview:

  This course introduces students to the fundamentals of programming using the Python programming language. Python is a versatile and beginner-friendly language widely used in various fields such as web development, data science, and scientific computing. The course covers basic programming concepts such as variables, data types, control structures, functions, and object-oriented programming.

Topics include file handling, working with databases, and using Python libraries for tasks such as data manipulation and visualization. By the end of the course, students will be able to write Python programs to solve a variety of computational problems and be prepared to pursue more advanced topics in Python programming.

11. Course objective:

- **Introduction to Programming**: Introduce students to the fundamentals of programming using Python as a beginner-friendly language.
- **Programming Concepts**: Teach core programming concepts such as variables, data types, control structures, functions, and object-oriented programming.
- **Problem Solving**: Develop problem-solving skills by working on coding exercises and projects that require algorithmic thinking.
- **Python Syntax and Semantics**: Familiarize students with the syntax and semantics of Python, including its unique features such as list comprehensions and lambda functions.
- **Data Structures and Algorithms**: Introduce common data structures (lists, dictionaries, etc.) and algorithms (sorting, searching, etc.) implemented in Python.

12. Student's requirements

   Software: Python 3.x, PyCharm distribution (recommended)

13. Forms of teaching

   I give hard copy of My lecture notes to students before coming lecturer time. first I remember students about previous lecture, and then I start new lecture. At the end of the lecture give a homework for the next lecture. During this proses I am use presentation and whiteboard.

14. Assessment scheme

1. *Midterm:* 20%, *Seminar*: 10%, *HW*: 10%: *Quiz*: 10%
2. *Final Exam: Theoretical:* 50% .

15. Student learning outcome:

- **Ability to Write Python Code**: Students will be able to write Python programs to solve a variety of computational problems.
- **Understanding of Basic Programming Concepts**: Students will demonstrate an understanding of basic programming concepts such as variables, data types, control structures, functions, and object-oriented programming.
- **Problem-Solving Skills**: Students will be able to analyze problems, develop algorithms, and implement solutions in Python.
- **Python Language Proficiency**: Students will demonstrate proficiency in using Python language features, libraries, and tools for programming tasks.
- **Software Development Practices**: Students will demonstrate an understanding of software development practices such as version control, debugging, and testing.
- **Data Structures and Algorithms**: Students will demonstrate an understanding of common data structures and algorithms and be able to implement them in Python.
- **Collaboration and Teamwork**: Students will demonstrate the ability to work effectively in teams, including sharing code, giving and receiving feedback, and resolving conflicts.
- **Preparation for Advanced Study**: Students will be prepared to pursue further study or careers in fields that require Python programming skills, such as data science, machine learning, or web development.

16. Course Reading List and References:

- Python, R., 2015. Python Basics: A Practical Introduction to Python 3.
- Heinold, B., 2021. A practical introduction to Python programming.
- Dowling, B., 2010. An introduction to Python for absolute beginners.
- Harrington, A.N., 2009. Hands-On Python A Tutorial Introduction for Beginners Python 3.1 Version. *Computer Science Department, Loyola University Chicago*.
- https://www.python.org/
- https://www.geeksforgeeks.org/python-programming-language/?ref=lbp
- https://www.w3resource.com/python/python-tutorial.php
- https://codesolid.com/category-python/

17. The Topics:	**Lecturer's name**

1. **Introduction to Python**
   • Installation and setup of Python environment
   • Basic concepts: variables, data types, operators, expressions, statements
2. **Data Structures**
   • Lists, tuples, dictionaries, sets
   • Operations on data structures: accessing, updating, deleting
3. **Control Structures**
   • Conditional statements: if, elif, else
   • Loops: for, while
   • Exception handling: try, except, finally
4. **Functions**
   • Defining functions
   • Parameters and arguments
   • Return statements
   • Scope and lifetime of variables
5. **Modules and Packages**
   • Importing modules
   • Creating and using packages
   • Standard library overview
6. **Object-Oriented Programming (OOP)**
   • Classes and objects
   • Inheritance
   • Polymorphism
   • Encapsulation
7. **Introduction to NumPy**
   • Install NumPy in PyCharm
	This Column are not applicable because timetables of holidays will change that is I cannot Determine a week by week review of the topics.

- NumPy Array Creation
- Numpy Array Indexing
- Basic Array Operations and Binary Operators
- Numpy Mathematical Function
- Numpy Linear Algebra

8. **Introduction to pyplot**
    - Install matplotlib in PyCharm
    - Formatting the style of your plot
    - Figure class and Axes Class
    - 3D Plots

18. Practical Topics (If there is any)

	This Column are not applicable because timetables of holidays will change that is I cannot Determine a week by week review of the topics.

19. Examinations:

20. Extra notes:

21. Peer review