# BASIC COURSE OUTLINE MODEL

## SCHOOL OF ENGINEERING & TECHNOLOGY

### DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

---

**Course Code**: ITEC322 **Credit Hour(s)**: **3**

**Webpage: vcampus.central.edu.gh** **Course Title**: Python Programming

---

**Course Lecturer: Mr. Charles Fomevor** **Room: C2 or E-40X**

**Email Address: cfomevor@central.edu.gh** **Tel no.:024-300 2308**

**Office Hours: Mondays, Tuesdays and Thursdays (10am-2pm)**

---

## COURSE OBJECTIVE

This course aims at providing working knowledge of Python to equip students with practical skills in areas like software engineering, data science, AI, and systems analysis.

## COURSE DESCRIPTION

This 3-hour credit course will equip students with problem-solving methods and algorithm development using the Python language. Students will develop object-oriented concepts applicable in areas such as web development, data science, software engineering, AI, etc. This includes data abstraction, debugging, library functions, control structures, passing parameters after a review of foundational concepts.

## LEARNING OUTCOMES

At the end of this course, students should be able to:

- recap related previous programming basics
- use varying data types and control structures
- understand inheritance and polymorphism among classes
- apply object-oriented programming concepts
- implement code for various scenarios in problem-solving

## INSTRUCTIONAL METHODS

Instructional approaches to be used during the course (e.g., lectures, presentations and assigned readings). **Note that attendance is also a requirement.**

## REQUIRED COURSE MATERIALS AND READINGS

*Guttag, J.*

Stewart, A. (2017). *Python Programming. Python Programming for Beginners, Python Programming for Intermediates.* North Charleston, South Carolina: CreateSpace Independent Publishing Platform.

Thompson, J. (2016). *PYTHON PYTHON'S COMPANION, A STEP BY STEP GUIDE FOR BEGINNERS TO START CODING.*

Yates, J. (2016). *Python Practical Python Programming For Beginners and Experts.* CreateSpace Independent Publishing Platform.

## EVALUATION

| | |
|---|---|
| Quizzes, Mid-Semester, Assignments etc | 40% |
| End of Semester Exams | 60% |
| Total | 100% |

## COMMIT TO ACADEMIC INTEGRITY

Students in the department are expected to maintain **high degrees of professionalism, commitment to active learning, participation and academic integrity every time**.

## ACADEMIC DISHONESTY

Please note that students involved in academic dishonesty will receive a **ZERO** mark on the particular component in which the infraction occurred and a notation of academic dishonesty in the departmental office. This may also reflect on references written by the department.
**It is the student's responsibility to understand what constitutes academic dishonesty.**

## MISSED EXAMS / TESTS / ASSIGNMENTS

**Assignment Submission**: Assignments must be received on the due date specified for the assignment.

**Lateness Penalty: Assignments received later than the due date will be penalized.** Exceptions to the lateness penalty for valid reasons such as illness, etc., may be entertained by the Lecturer but will require supporting documentation (e.g., a doctor's letter).

**Missed Tests:** Students with a documented reason for missing a course test, such as illness, which is confirmed by supporting documentation (e.g., doctor's letter) will be handled by the course lecturer.

## WEEK-BY-WEEK COURSE SCHEDULE / ORGANISER:

| Week | Topic | Activities | Due Date |
|---|---|---|---|
| 1 | **Introduction**<br>Programs & Algorithms,<br>Introduction to Python, Algorithm,<br>Basic debugging | Lectures begin | Week 1 |
| 2 | **Fundamentals of Python Data**<br>Variables, Expressions, and Statements,<br>Values and Data Types | Lecture<br>Reading | Week 2<br>**Quiz 1** to be held on week 3 |
| 3 | **Debugging in Python**<br>Avoiding debugging<br>Helpful tips for debugging<br>Identifying error messages | Lecture<br>Reading<br>**Quiz 1** | Week 3<br>Quiz 1 |
| 4-5 | **Program flow and conditionals**<br>Loop types<br>Iteration and the range functions | Lecture<br>Reading<br>**Quiz 2** | Week 4 – Week 5<br><br>Week 5 – Quiz 2 |
| 6 | **Functions in Python**<br>Introduction to functions<br>Local variables and parameters<br>Nesting functions<br>Program development with functions | Lecture<br>Reading<br>**Assignment 1 (mini-project)** | Week 6<br><br>Week 6 – Exercises |
| 7 | **Selection**<br>Expressions using Boolean values<br>Precedence with operators<br>Conditionals and execution | Lecture<br>Reading | Week 7<br><br>Week7<br>Submit Assignment 1 |
| 8-9 | **Further Iterations**<br>For and While Loops with sentinels<br>Useful algorithms with iterations | Lecture<br>Reading<br>**Assignment 2** | Week 8 – Week 9<br><br>Week 8 – Exercises |
| 10-11 | **Strings**<br>Data Type collections,<br>Indexing<br>String methods and slicing | Lecture<br>Reading | Week 10 - 11<br>Mid Semester<br><br>Week 10 |

| Week | Topic | Activities | Due Date |
|---|---|---|---|
| | | | Submit Assignment 2 |
| 12 | **Lists, Files, Dictionaries**<br>Concatenation, repetition, deletion<br>Objects and references<br>Lists as parameters and return<br>Principles of using exception | Lecture<br>Reading | Week 12 |
| 13 | **Classes and Objects**<br>User-defined classes<br>Constructors and other methods<br>Objects as arguments and parameters | Lectures end<br><br>Review of Semester Work | Week 13 |
| 14 | | Revision Week | |
| 15 | | Exams begin | |
| | | Exams end / vacation | |