COL’s Blended Learning Design Template

**Programme:** Bachelor of Science in Computer Science

**Course title**: Generic Programming using Python

**Course facilitator:** Juma Kilwake

**Course description:** Python is a language with a simple syntax, and a powerful set of libraries. It is an interpreted language, with a rich programming environment, including a robust debugger and profiler. While it is easy for beginners to learn, it is widely used in many scientific areas for data exploration. This course is an introduction to the Python programming language for students with prior programming experience. We cover data types, control flow, object-oriented programming. Python is a versatile programming language, suitable for projects ranging from small scripts to large systems. Students will explore the large standard library of Python 3, which supports many common programming tasks.

**Learning outcomes:** Upon successfully completing this course, learners are expected to be able to:

* Identify/characterize/define a problem
* Design a program to solve the problem
* Use Python scripting elements such as variables and flow control structures
* Create Python functions to facilitate code reuse
* Use Python to read and write files
* Create robust code by handling errors and exceptions properly
* Use the Python standard library
* Implement Python's object-oriented features

**Course structure:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Structure**  **by Unit** | **Learning Outcomes** | **Assessments: F2F/Moodle-enabled** | **Learning Activities: F2F/Moodle-enabled** | **Learning Content:**  **F2F/Moodle-enabled** | | **Facilitating Online** |
| **Self-created/ Web Resources** | **Supportive OER with TASL Attribution** |
| **Week 1 :** Python Basic Data Types, Operators and Variables | LO 1  LO 2  LO 3  LO 4 | FA 1 (LO 1)  FA 2 (LO 1)  SA 1 (LO 1 & 2)  FA 3 LO 3  FA 4 LO 4  SA 2 (LO 3 & 4) |  |  |  |  |
| **Week 2:** Lists and strings |  |  |  |  |  |  |
| **Week 3:** Dictionaries |  |  |  |  |  |  |
| **Week 4:** Conditional statements: boolean/if/elif/else, compound conditionals |  |  |  |  |  |  |
| **Week 5:** Loops: while loop, for loop |  |  |  |  |  |  |
| **Week 6:** Loops: while loop, for loop |  |  |  |  |  |  |
| **Week 7:** Methods and Functions, recursive functions |  |  |  |  |  |  |
| **Week 8:** Reading from and writing to files, |  |  |  |  |  |  |
| **Week 9:** Errors and Exceptions |  |  |  |  |  |  |
| **Week 10:** Object Oriented Programming: classes, objects, encapsulation |  |  |  |  |  |  |
| **Week 11:** Inheritance, Polymorphism |  |  |  |  |  |  |
| **Week 12:** Modules and Packages |  |  |  |  |  |  |
| **Week 13:** Decorators and Generators |  |  |  |  |  |  |
| **Week 14:** Working with numpy |  |  |  |  |  |  |

**Notes:**

1. Learning outcomes: Statements that specify what learners will be able to do as a result of learning

2. Assessments: Formative assessment (FA), Summative Assessment (SA), Peer Assessment or Self-assessment and Tools (MCQ, essay-type questions, project work, etc.)

3. Learning activities: active learning, collaborative learning, constructive learning, social learning

4. Learning content: Print – textbooks, downloadable PDF/PPT/Word documents; Multimedia – lecture videos, animations, images, YouTube/Vimeo/Khan Academy videos, OER, etc.

5. Facilitating online:

1. Create a course introductory video (about this course, learning outcomes, course outline, learning activities and assessments, grading policy, expected participation) and a unit introductory video, if required.
2. Share course handout/session plan/academic plan.
3. Send introductory email to students one week prior to course start date with pre-course preparatory activities.
4. Share your contact details and times, channels of communication and turnaround times for grading assignments and responding to students’ queries.
5. Provide contact details of technical support staff for troubleshooting login issues.
6. Create FAQ on how to access and navigate the course site and learning resources, and how to submit learning activities and assessments.
7. Send weekly email communication to students to wrap up a unit/topic and introduce the next unit/topic.
8. Engage learners in interaction with peers and faculty, either synchronous or asynchronous.
9. Provide learning support through discussion forums. Create forums for:
   * introductions
   * course announcements (to establish online course presence)
   * posting general queries and seeking learning support (encourage students to provide peer support)
   * posting learning reflections (encourage students to rate their peers’ reflections)
10. Engage learners in self-reflection, knowledge sharing and co-creation, recognising learners’ contributions with badges.
11. Track student progress — course participation, completion of activities and assessments — and alert non-participants.
12. Create rubrics for maintaining transparency in grading.
13. Provide timely and constructive feedback/feed-forward to improve learning.
14. Seek students’ feedback on course and self.