Semester Plan: Weeks 11–16

WEEK 11 **Functions and dictionaries**
Finish functions. Start creating, looking up, and updating dictionary entries.

WEEK 12 **Dictionaries and classes**
Finish dictionary practice. Create objects with __init__, self, and methods.

WEEK 13 **Object-oriented programming**
Work with multiple objects, organize methods, and test object behavior.

WEEK 14 **Files, JSON, and project kickoff**
Two periods: text files, JSON save/load, and one missing-file exception.
Two periods: define a small first version and start coding.

WEEK 15 **Project development**
Finish core features, test the program, and prepare a short walkthrough.

WEEK 16 **Code demonstrations only**
Run your program and explain your code in class or individually.

How Your Final Grade Is Calculated

Coursework contributes 50%; the independent final project contributes 50%.



PROGRAMMING EXERCISES

Five assignments completed during the semester.

CLASS PARTICIPATION

Attendance, discussions, coding, and class tasks.

FINAL PROJECT

An independently developed Python program.

TOTAL: 30% + 20% + 50% = 100%

Coursework: 50% of Your Final Grade

Two components: programming exercises (30%) and class participation (20%).

30% Programming Exercises

Five exercises throughout the semester.

Each exercise is graded on:

- Program correctness
- Completeness
- Code quality and style
- Submission on time

20% Class Participation

Participation is assessed through:

- Attendance
- Contributions to discussions
- Participation in coding activities
- Completion of in-class tasks

Final Project: What You Need to Do

The independent Python programming project is worth 50% of your final grade.

Design and implement your own Python program.

Your project should demonstrate your understanding of:

01 **Data structures**

02 **Functions**

03 **Object-oriented programming**

Be prepared to clearly explain or demonstrate how your program works.

How the Final Project Is Assessed

The project is evaluated based on both the program and its explanation or demonstration.

1 Program design

Quality of your overall solution

2 Code structure

Clear, organized code

3 Use of Python features

Appropriate application of Python

4 Functionality

Your program works as intended

5 Explanation / demo

Clear presentation of your work