

Concepts of Programming Languages
Fall 2025-2026
Tasks Ideas

❖ **Task Requirements**

You must implement one of the following tasks twice:

1. Using a **pure functional programming paradigm**
2. Using an **imperative programming paradigm**

Important Note: The implementation must be done in one of the following languages: Python, JavaScript, OZ, or C#.

❖ **Task Motivation**

The goal of implementing tasks using both functional and imperative paradigms is to help you:

- Understand the core differences between these paradigms.
- Explore their strengths and weaknesses.
- Develop a deeper appreciation for how different paradigms shape problem-solving in programming.

❖ **Task Grading**

- The imperative paradigm is graded out of 5.
- The functional paradigm is graded out of 5.

Tasks ideas

1. Functional Data Processing Pipeline

This task involves building a system to process datasets. You must load data from sources such as CSV, JSON, or databases.

Key Features

- Handle missing data (e.g., fill with defaults or remove).
- Standardize formats (e.g., dates, numerical precision).
- Data Transformation:
 - Filter rows based on conditions (e.g., Sales > 1000).
 - Compute new columns (e.g., Sales Growth).
 - Aggregate data by key (e.g., total sales per region).
- Data Analysis:
 - Perform operations such as:
 - Statistical summaries (mean, median, variance).
 - Trend analysis or correlation between variables.
- Data Visualization (Optional):
 - Produce charts such as bar graphs or line charts.
- Output Results:
 - Save processed data to files (e.g., a clean CSV) or display summaries to the console.

2. Sudoku Game

Develop an AI system to solve Sudoku Puzzle. The standard version of Sudoku consists of a 9×9 square grid containing 81 cells. The grid is subdivided into nine 3×3 blocks. Some of the 81 cells are filled in with numbers from the set $\{1,2,3,4,5,6,7,8,9\}$. These filled-in cells are called givens. The goal is to fill in the whole grid using the nine digits so that each row, each column, and each block contains each number exactly once.

3. N Queen Puzzle

Develop an AI agent to solve an N Queen Puzzle. The N Queens Puzzle is a classic problem in chess and computer science that involves placing eight chess queens on a standard 8×8 chessboard so that no two queens attack each other.

The objective is to arrange all eight queens on the board in a way that none of them can capture any of the others using standard chess queen moves.

4. Conway's Game of Life

It is a **simulation** game that shows how simple rules can produce complex patterns over time. The game is played on a 2D grid of cells. Each cell can be either alive or dead.

The state of the grid evolves generation by generation based on simple rules.

Rules of the Game

For each cell in the grid:

Underpopulation: A live cell with fewer than 2 live neighbors dies.

Survival: A live cell with 2 or 3 live neighbors stays alive.

Overpopulation: A live cell with more than 3 live neighbors dies.

Reproduction: A dead cell with exactly 3 live neighbors becomes alive.

Neighbors: Each cell has 8 neighbors (horizontal, vertical, and diagonal).

5. Sliding Tile Puzzle

The 8-puzzle is a classic problem in artificial intelligence and problem-solving. It consists of a 3×3 grid with 8 numbered tiles and one empty space. The goal is to move the tiles around until they are in a specific target order. Create an AI agent that solves the 8-puzzle.

6. Simple Simulation: Animals Moving on a Grid

Simulates an ecosystem where creatures move, eat, survive, and reproduce on a grid-based environment. This is a grid-based simulation where animals move around a 2D world. Each cell in the grid can be empty or contain an animal. You have to implement simple rules to move animals and simulate interactions like eating, or avoiding obstacles... etc.