**Python Paradigm**

**Procedural Organisation:** The Python code is structured primarily by functions, which are arranged in a procedural fashion. It promotes a series of procedural steps above clear class structures.

**Functional Decomposition:** Using functions like `check\_winnings’, `print\_slot\_machine’, and `generate\_random\_number’, the primary functionality is achieved. These functions encourage a modular and deconstructed structure by encapsulating particular responsibilities.

**Global Variables:**

The use of global constants (`ONEPOUND\_BET`, `ROWS`, `COLS`, etc.) and dictionaries (`symbol\_count`, `symbol\_value`, etc.) is consistent with procedural programming, where data is often managed through global variables.

**Linear Execution:**

The program's control flow is based on a linear execution model. The game's growth is controlled by user input inside of a loop, with the {main` function serving as the entrance point.

**Data-Oriented Approach:**

The program utilises dictionaries and lists to organize and manipulate data. For instance, `symbol\_count` and `symbol\_value` dictionaries manage the counts and values of symbols, representing a data-centric approach.

**Modularity Through Functions:**

The code promotes modularity using functions. Each function serves a specific purpose, contributing to the overall functionality of the slot machine game.

**Sequential Flow:**

The program's logic follows a sequential flow, moving through different functions and operations based on user choices and random outcomes.

**No Explicit Use of Classes:**

In contrast to an object-oriented approach, the code does not explicitly define classes or involve object instances. The focus is on procedures and functions.

The Python solution effectively incorporates procedural programming features, with an emphasis on functions, global variables, and a step-by-step execution model. This procedural style promotes readability, modularity, and a clear sequence of operations within the context of the slot machine game.