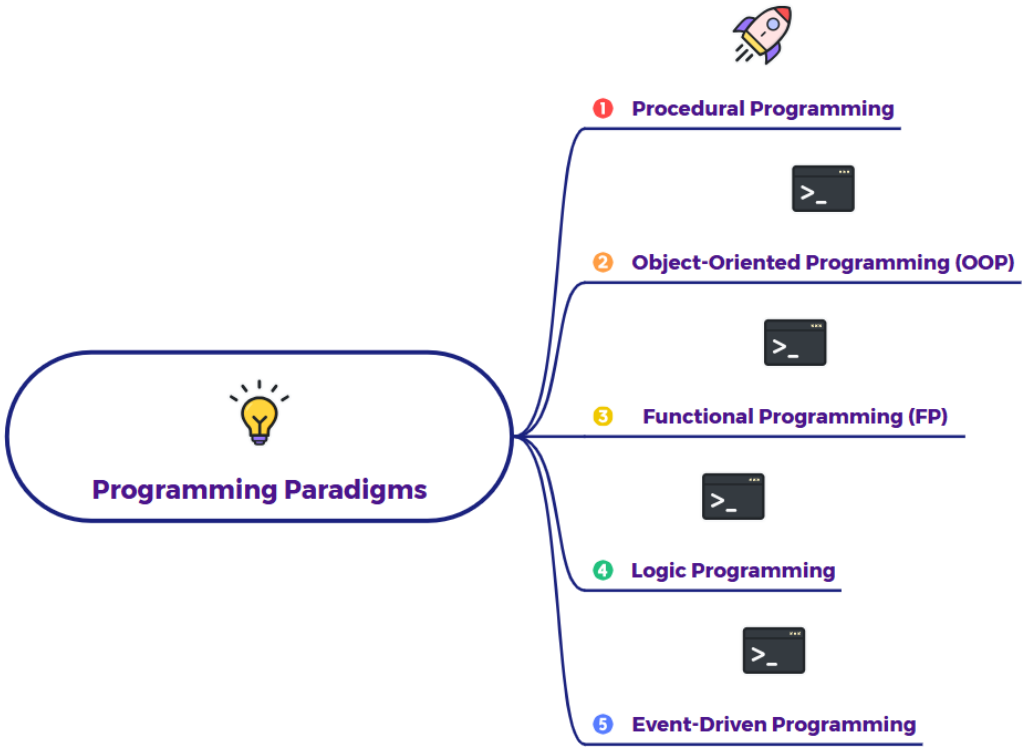
Explain Procedural Programming with an example



Imagine you're trying to teach someone how to bake a cake using procedural programming principles.

The "Procedural Cake Baking" Way:

You wouldn't just say, "Bake a cake." Instead, you'd give them a **very specific, step-by-step list of instructions**, one after the other. Each step tells them *exactly what to do* to the ingredients (which are like your "data").

Here's your recipe (your "program"):

1. **Procedure: PreheatOven**
    - **Instruction 1:** Turn on the oven.
    - **Instruction 2:** Set temperature to 180°C.
    - **Instruction 3:** Wait until the oven light turns off.
2. **Procedure: PrepareIngredients**
    - **Instruction 1:** Get flour from the pantry.

- **Instruction 2:** Get sugar from the pantry.
- **Instruction 3:** Get eggs from the fridge.
- **Instruction 4:** Measure 2 cups of flour.
- **Instruction 5:** Measure 1 cup of sugar.
- **Instruction 6:** Crack 3 eggs into a bowl.

3. **Procedure: MixBatter**
    - **Instruction 1:** Put measured flour, sugar, and eggs into a large mixing bowl.
    - **Instruction 2:** Add 1/2 cup of milk.
    - **Instruction 3:** Add 1 teaspoon of baking powder.
    - **Instruction 4:** Stir everything until smooth.
4. **Procedure: BakeCake**
    - **Instruction 1:** Pour batter into a greased cake tin.
    - **Instruction 2:** Place tin carefully into the preheated oven.
    - **Instruction 3:** Set timer for 30 minutes.
    - **Instruction 4:** Wait for timer to ring.
    - **Instruction 5:** Carefully remove cake from oven.

Key Points of Procedural Programming in this Example:

- **Steps/Instructions (Procedures):** Every action is a clear, distinct instruction. You organize these instructions into logical groups (like PreheatOven, MixBatter), which are called "procedures" or "functions."
- **Order Matters:** You have to follow the steps in the exact sequence. You can't bake the cake before mixing the batter, or mix the batter before getting ingredients.
- **Working on Data:** All the instructions (PreheatOven, MixBatter, etc.) operate directly on the "ingredients" (flour, sugar, eggs) and the "oven" or "bowl" (which are like pieces of data or parts of the program's "state"). The ingredients are changed (mixed, cooked) by the procedures.

- **Global Access (Sometimes):** Imagine all your ingredients are just sitting on the kitchen counter, available to any procedure at any time. This is like "global data" – any function can access and change it. (This can sometimes get messy if too many procedures are fiddling with the same ingredients!).
- **Recipe Focus:** The main focus is on the *recipe* itself – the series of actions needed to achieve the goal (a baked cake).

In short: Procedural programming is like writing a detailed recipe where you tell the computer exactly what steps to perform, in what order, to achieve a specific result, operating directly on the available data.