**✅ Phases of Your Pebble Language Project**

**📂 Sample Pebble Code (test.peb)**

let a = 10;

let b = 20;

display(a + b);

**🔷 1. Pebble Source Code (.peb file)**

* User writes code in .peb file.
* **Example:** test.peb
* Contains high-level Pebble syntax.

**🔷 2. Lexical Analyzer (Lex)**

* Reads .peb file and breaks code into **tokens**.
* Recognizes:
  + Keywords → let, display
  + Identifiers → a, b
  + Numbers → 10, 20
  + Operators → =, +
  + Symbols → ;, (, )

**🔧 Tool:** Lex (pebble.l)

**🎯 Output:**

<TOKEN: let> <ID: a> <ASSIGN: => <NUM: 10> <SEMICOLON>

...

**🔷 3. Parser (Syntax + Semantic Checker)**

* Checks if token sequence follows correct grammar.
* Also performs semantic checks:
  + Variable declared or not
  + Type consistency

**🔧 Tool:** YACC or custom parser (pebble.y or manually written) **✅ Output:** Success → generates intermediate representation OR  
Error → “Variable not declared”, “Missing semicolon”, etc.

**🔷 4. Compilation Engine (Translator)**

* Converts Pebble syntax to equivalent **C code**.
* Example Pebble:

let a = 10;

display(a);

⬇️ becomes:

int a = 10;

printf("%d\n", a);

**🔧 File:** pebble\_compiler.c or similar

**🎯 Output:** output.c

**🔷 5. GCC Compiler**

* Takes the generated output.c
* Compiles it to machine code (a.out or output.exe)

**🔧 Tool:** GCC (gcc output.c -o run)

**🎯 Output:** Executable file

**🔷 6. Execution**

* Run the compiled file to see output

**Example Output:**

30

**🔁 Pebble Project Phase Flow (Summary Table)**

| **Phase** | **Tool/File** | **Function** | **Output** |
| --- | --- | --- | --- |
| Pebble Code Input | test.peb | Source written in Pebble syntax | Raw Pebble code |
| Lexical Analysis | pebble.l (Lex) | Break into tokens | Token stream |
| Parsing (Syntax & Sem) | pebble.y / parser | Grammar & type check | Parse Tree or IR |
| Compilation Engine | pebble\_compiler.c | Translate Pebble to C | output.c |
| Compilation (GCC) | gcc | Compile translated code | Executable file |
| Execution | ./run | Run the program | Final Output |

**📊 Diagram Idea for Report/Slides**

You can draw this flow:

┌──────────────┐

│ test.peb │

└─────┬────────┘

↓

┌────────────────────┐

│ Lexical Analyzer │

│ (pebble.l) │

└─────┬──────────────┘

↓

┌────────────────────┐

│ Syntax & Semantic │

│ Checker (Parser) │

└─────┬──────────────┘

↓

┌────────────────────┐

│ Compilation Engine │

│ (Pebble to C) │

└─────┬──────────────┘

↓

┌────────────────────┐

│ GCC Compiler │

└─────┬──────────────┘

↓

┌────────────────────┐

│ Executable Output │

└────────────────────┘

**✅ Want to Stand Out?**

You can add these:

* ✅ "Errors shown in each phase" demo
* ✅ Side-by-side Pebble vs C code examples
* ✅ Explain why you chose this compiler approach (portability, simplicity)