In this assignment, you'll utilize Node.js and TypeScript to construct a useful Command-Line Interface (CLI) using the commander.js library. The objective of this CLI tool is to generate boilerplate code for challenges.

Tasks

```
npm install typescript --save-dev
          +                       =
npm install @types/node --save-dev
```

```
npm i -D typescript @types/node
```

1. Initial setup

Before getting started, ensure you follow these steps to set up the project and install all the necessary dependencies:

- Create a Node.js application named **codecla-cli**.
- Install the TypeScript compiler: **npm install typescript**.
- Install Node.js types: **npm install @types/node --save-dev**.
- Install the CommanderJS library.
- Install the inquirer library and its type definitions.

2. CLI Development

The CLI should accept these arguments:

- Function name.
- Programming language.
- List of function inputs.

With this information, the CLI will generate the boilerplate code for a coding challenge.

Note: Boilerplate, are sections of code that are repeated in multiple places with little to no variation

The CLI usage would be as follows:

```
code-cli -n function_name -l javascript -i a,b
```

- **-n** to specify the function name.
- **-l** to specify the programming language.
- **-i** to specify the list of function inputs (comma-seperated).

There are only two supported languages: python and javascript. After the boilerplate has been generated, it should be stored in a file with the same name as the function name and it should be given the proper file extension (.py for python and .js for js).

Python boilerplate example

```python
def functionName(a):
    # Your code here
    return
```

JavaScript boilerplate example

```javascript
function functionName(a) {
  // Your code here
  return;
}
```

- Implement a Commander.js command to generate boilerplate code.
- Before generating the boilerplate, prompt the user for confirmation using the Inquirer library, with a message like "Generate boilerplate code for the function solution in the specified programming language?"
- Save the generated code in a file named after the function, with the appropriate file extension corresponding to the chosen programming language.
- Ensure that types are added for all objects used in the CLI.

Demo

We've prepared a demonstration showcasing how the CLI should appear in action. You can watch it for reference.