**1. Number Guessing Game**

* **Description**: Create a simple number guessing game where the computer generates a random number, and the player has to guess it within a certain number of attempts.
* **Key Concepts**: random module, conditionals, loops, user input.

**2. Simple Calculator**

* **Description**: Build a basic calculator that can perform addition, subtraction, multiplication, and division based on user input.
* **Key Concepts**: Functions, user input, conditionals.

**3. To-Do List**

* **Description**: Develop a to-do list where users can add tasks, mark them as complete, and delete them. It can be a text-based program, and later you could add a simple graphical interface using libraries like Tkinter.
* **Key Concepts**: Lists, functions, loops, file handling (to save tasks).

**4. Rock, Paper, Scissors Game**

* **Description**: Create a simple text-based game where the player chooses between rock, paper, or scissors, and the computer randomly selects one. Then, the program determines the winner.
* **Key Concepts**: random module, conditionals, user input.

**5. Hangman Game**

* **Description**: Implement the classic Hangman game where the player guesses letters to reveal a word before running out of attempts.
* **Key Concepts**: Strings, loops, conditionals, lists.

**6. Simple Countdown Timer**

* **Description**: Create a countdown timer that takes user input for the time and counts down to zero, printing a message when the time is up.
* **Key Concepts**: time module, loops, user input.

**7. Unit Converter**

* **Description**: Build a simple unit converter that converts units such as temperature (Celsius to Fahrenheit), distance (kilometers to miles), or weight (kilograms to pounds).
* **Key Concepts**: Functions, user input, conditionals.

**8. Simple Contact Book**

* **Description**: Create a simple program that allows users to store contacts with names, phone numbers, and emails. You can save the data in a file and allow for adding, viewing, and deleting contacts.
* **Key Concepts**: Dictionaries, file handling, user input.

**9. Palindrome Checker**

* **Description**: Write a program that checks if a word or phrase is a palindrome (a word that reads the same forwards and backwards).
* **Key Concepts**: Strings, loops, conditionals.

**10. Fibonacci Sequence**

* **Description**: Create a program that generates the Fibonacci sequence up to a certain number of terms, either iteratively or recursively.
* **Key Concepts**: Loops, recursion, functions.

**11. Simple Alarm Clock**

* **Description**: Build a simple alarm clock that lets the user set a time and then plays a sound or prints a message when the time arrives.
* **Key Concepts**: time module, loops, user input.

**12. Number to Words Converter**

* **Description**: Write a program that converts a number (e.g., 123) into its word equivalent ("one hundred twenty-three").
* **Key Concepts**: Strings, conditionals, functions.

**13. Basic Web Scraper**

* **Description**: Create a basic web scraper using libraries like requests and BeautifulSoup to extract data from a website, such as news headlines or stock prices.
* **Key Concepts**: Web scraping, requests, BeautifulSoup.

**14. Basic Text-Based Adventure Game**

* **Description**: Create a text-based adventure game where players can navigate through rooms and make choices that affect the outcome of the game.
* **Key Concepts**: Functions, conditionals, loops, dictionaries.

**15. Tic-Tac-Toe Game**

* **Description**: Build a console-based version of the classic Tic-Tac-Toe game that allows two players to play on the same computer.
* **Key Concepts**: Lists, loops, conditionals, user input.