20CYS312 - Principles of Programming Languages
Exploring Programming Paradigms

Assignment-01

Presented by Aishwarya GS
CB.EN.U4CYS21004
TIFAC-CORE in Cyber Security
Amrita Vishwa Vidyapeetham, Coimbatore Campus

Feb 2024

AMRITA
VISHWA VIDYAPEETHAM

Outline

Understanding Declarative Programming

- Emphasis on specifying "what" rather than "how"
- Expressing desired outcomes without step-by-step instructions
- Characteristics:
  - Descriptive Syntax
  - Abstraction
  - Expressive Constructs
  - Immutability
  - Data-Driven
  - Functional Programming Concepts

Descriptive Syntax in Declarative Languages

- Descriptive syntax expresses desired outcomes
- Avoids detailing step-by-step procedures
- Example: SQL, HTML

Abstraction in Declarative Programming

- Focus on "what" rather than "how"
- High-level abstractions simplify development
- Example: Declarative UI frameworks

Data-Driven and Functional Concepts in Declarative Paradigm

- Emphasis on data-driven programming
- Integration of functional programming concepts:
- Pure functions
- Higher-order functions
- Function composition
- Enhancing expressiveness in the declarative style

Introduction to SQL in Declarative Paradigm

- SQL in Declarative Paradigm
- Database interaction by specifying desired results
- SQL handles behind-the-scenes work

Declarative Constructs in SQL

- Core declarative constructs: SELECT, INSERT, UPDATE, DELETE
- Specifying desired data without detailing steps
- Example SQL queries:
  - SELECT column1, column2 FROM table WHERE condition;
  - INSERT INTO table (column1, column2) VALUES (value1, value2);
  - UPDATE table SET column1 = value1 WHERE condition;
  - DELETE FROM table WHERE condition;

Focusing on "What" in SQL

- Contrast with imperative paradigm
- Emphasis on what data is needed
- System handles details of retrieval, insertion, updating, and deletion

Declarative Structure with DDL

- DDL statements define and modify database structure
- Example: CREATE TABLE statement
- Showcase of SQL's declarative nature in defining database structure

Event-Driven Programming Overview

- Centered around events and responses
- Flow of the program determined by events
- Characteristics of the event-driven paradigm

Core Features of Event-Driven Programming

- Asynchronous Operations: Allows handling multiple events simultaneously
- Callback Functions: Executed in response to specific events
- Event Listeners: Mechanisms for attaching to respond to events
- Event Emitters: Constructs triggering events, notifying listeners
- Observer Pattern: Objects register interest and are notified of changes
- Custom Event Creation: Facilitates communication between program parts

Asynchronous Operations in Event-Driven Programming

- Handling events without blocking the main program flow
- Importance of asynchronous operations
- Callback functions as essential for event handling

Dynamic and Interactive Software Development

- Dynamic and interactive dimension
- Modular and flexible approach
- Effective communication between program components
- Adapting to dynamic inputs and environmental changes

JavaScript and Event-Driven Programming

- Inherent event-driven nature
- User interactions and system events trigger code execution
- Key aspect in web development

Event Handling in JavaScript

- Event Listeners: "Listen" for specific events
- Example: Clicks, key presses, mouse movements
- Event Objects: Contain information about the event; Passed as arguments to event handler functions
- Example: Logging target element on button click

```
const button = document.getElementById('myButton');
button.addEventListener('click', function(event)
console.log('Button clicked! Target:', event.target);
);
```

Callback Functions and Asynchronous Operations

- Callback Functions: Executed later, often in response to an event
- Example: Handling click events with a callback function
- Asynchronous Operations: Promises and async/await syntax for managing asynchronous code
- Example: Fetching data from a server asynchronously

```
async function fetchData()
const response = await
fetch('https://api.example.com/data');
const data = await response.json();
console.log('Data fetched:', data);
fetchData();
```

Custom and DOM Events in JavaScript

- Custom Events: Creation and dispatching for communication between components
- DOM Events: Heavily event-driven, especially with Document Object Model (DOM) events
- Example: Handling button clicks in the browser environment

```
const customEvent = new Event('customEvent');
document.addEventListener('customEvent', function()
console.log('Custom event triggered!');
);
document.dispatchEvent(customEvent);
```

- Promises for Asynchronous Handling: Promises for cleaner handling of asynchronous operations
- Example: Handling asynchronous operations with promises

```
function asyncOperation()
return new Promise((resolve, reject) =>
// Asynchronous code here
resolve('Operation complete!');
);
asyncOperation().then(result => console.log(result));
```

Comparisons and Contrasts of Paradigms

- Comparisons
  - Declarative
    - Focus on "what" needs to be done, abstracting away details.
    - Well-suited for describing data, relationships, configurations, and transformations.
  - Event-Driven
    - Focus on responding to events, specifying how the program behaves.
    - Ideal for building interactive interfaces, handling asynchronous tasks, and responding to real-time events.
- Contrasts
  - Declarative Paradigm: Emphasizes expressing desired outcomes without prescribing explicit steps.
  - Event-Driven Paradigm: Focuses on responding to events, frequently using asynchronous programming for real-time interactions.

References

- https://www.geeksforgeeks.org/introduction-of-programming-paradigms/
- https://www.techtarget.com/searchitoperations/definition/declarative-programming
- https://www.studysmarter.co.uk/explanations/computer-science/computer-programming/event-driven-programming/
- https://habtesoft.medium.com/event-driven-programming-how-does-it-relate-to-node-js-2885af9b87c0: :text=js
- https://medium.com/@datasciencenexus/sql-theory-sql-as-a-declarative-language-9703912bd01c
- https://en.wikipedia.org/wiki/Comparisonofprogrammingparadigms
- https://www.decipherzone.com/blog-detail/programming-paradigms
- https://coderpad.io/blog/development/addeventlistener-javascript/
- https://www.sqlshack.com/learn-sql-sql-query-examples/
- https://www.oreilly.com/content/why-reactive/
- https://m.youtube.com/watch?v=E7Fbf7R3x6I
- http://conal.net/papers/