DIT 62205 Event Driven programming

Course Code	**DIT 62205**
Course Title	**Event Driven programming**
Lecture Hours	30
Practical Hours	30
Tutorial Hours	N/A
Contact Hours	60
Credit Units	4
Description	The course is designed to provide an introduction the paradigm of programming where the flow of the program is determined by events such as user actions, sensor outputs, or messages from other programs or threads. Students will learn how to design interactive applications that respond dynamically to user input, providing a foundation for building responsive and interactive software systems.
Audience Description	This is a course intended for second year students
Delivery Mode	Blended approach: both face-to-face and online
Course Objective	The course is meant to facilitate you to; • Introduce concepts of event-driven programming and its significance in modern software development. • Design interactive user interfaces and applications that effectively utilize event-driven programming techniques. • Implement event handlers to respond to various types of events, such as user interface interactions, system events, and custom events. • Apply event-driven programming concepts to create real-world applications, ranging from graphical user interfaces to interactive web applications

Expected Learning Outcomes	i. Compare the concepts of event-driven programming and its significance in modern software development. ii. Design interactive user interfaces and applications that effectively utilize event-driven programming techniques. iii. Implement event handlers to respond to various types of events, such as user interface interactions, system events, and custom events. iv. Apply event-driven programming concepts to create real-world applications, ranging from graphical user interfaces to interactive web applications
Brief Content Description	**Topic 1: Introduction to Event-Driven Programming** - Understanding the event-driven paradigm and its applications. - Differentiating between event-driven and procedural programming. - Exploring event loops and event queues. **Topic 2: GUI Programming with Events** - Introduction to Graphical User Interfaces (GUIs). - Implementing event handlers for GUI components (buttons, menus, etc.). - Creating interactive applications using a GUI library (e.g., Tkinter, JavaFX, or similar). **Topic 3: Handling User Interactions** - Capturing user input events (e.g., mouse clicks, keyboard presses). - Implementing responsive interfaces using event-driven techniques. - Creating dynamic UI components that adapt to user actions. **Topic 4: Event-Driven Web Development** - Introduction to event-driven programming in web development. - Implementing event handlers in JavaScript for DOM events. - Building interactive web applications using frameworks like React or Angular.

	Topic 5: Custom Events and Event Propagation - Creating custom events to handle specific application requirements. - Understanding event propagation and bubbling. - Managing event flow in complex applications. **Topic 6: Asynchronous Programming and Callbacks** - Exploring asynchronous operations and their role in event-driven systems. - Implementing callback functions to handle asynchronous events. - Utilizing Promises or async/await for managing asynchronous code. **Topic 7: Event-Driven Design Patterns** - Exploring common design patterns in event-driven programming (e.g., Observer, Pub-Sub). - Applying design patterns to solve specific programming challenges. **Topic 8: Error Handling and Debugging in Event-Driven Programs** - Handling errors and exceptions in event-driven applications. - Debugging techniques for event-driven codebases. - Best practices for ensuring robustness and reliability. **Topic 9: Event-Driven Networking** - Understanding event-driven programming in network applications. - Implementing socket programming with event-driven frameworks. **Topic 10: Real-World Applications and Case Studies** - Analyzing and building practical applications that demonstrate the power of event-driven programming. - Reviewing real-world examples of event-driven systems in action.

Teaching and Learning Methods	• Lectures • Lecturer guided practical sessions • Group and individual tasks
Learner Support Plan	• Student will have regular physical meetings with their instructors and the department administration. • Centralized communication channels will be established to enable instant communication which is all-inclusive. E.g. class emails, WhatsApp groups, Slack and GitHub. • Student project groups and mentorship program
Resources/ Teaching and learning materials	These will mainly be disseminated through the UMU E-learning system. They will include; • Pre-recorded lectures • Lecture handouts • Practical/Lab examples will be shared on the Git Repository • Case studies and Project activities • Open Educational Resources (OER) like; Academic papers/articles, E-Books
Course Assessment Plan	**Coursework:** 50% **EXAM:** 50%
Supporting Technologies	• Synchronous: Zoom, Google Meet • Asynchronous: Discussion Forum, • Other software: PowerPoint, GitHub and Git, Google Drive, Learning Management System etc.
Core texts	1. Flanagan, D. and Novak, G.M., 1998. Java-Script: The Definitive Guide. 2. Moore, A., 2021. *Python GUI Programming with Tkinter*. Packt Publishing. 3. Dea, C., Grunwald, G., Pereda, J., Phillips, S. and Heckler, M., 2017. *JavaFX 9 by Example*. Apress.

	4. Bertoli M. React Design Patterns and Best Practices. Packt Publishing Ltd; 2017