Republic of the Philippines

SULTAN KUDARAT STATE UNIVERSITY

Isulan, Sultan Kudarat

College of Computer Studies

S.Y. 2024-2025

IT223

Event Driven Programming

2nd Semester

School Year 2024 – 2025

Prepared by:

KRISTINE MAE H. AMPAS, MIT

UNIVERSITY VISION

A trailblazer in arts, science and technology in the region.

UNIVERSITY MISSION

The University shall primarily provide advance instruction and professional training in Science and Technology, Agriculture, Fisheries, Education and other related field of study. It shall undertake research and extension services, and provide progressive leadership in its area specialization.

UNIVERSITY GOAL

To produce graduates with excellent and dignity in arts, science na technology.

UNIVERSITY OBJECTIVES

a. Enhance competency development, commitment, professionalism, unity and true spirit of service for public accountability, transparency and delivery of quality services;
b. Provide relevant programs and professional trainings that will respond to the development needs of the region;
c. Strengthen local and international collaborations and partnership for borderless programs;
d. Develop a research culture among faculty and students;
e. Develop and promote environmentally-sound and market-driven knowledge and technologies at par with international standards;
f. Promote research-based information and technologies for sustainable development;
g. Enhance resource generation and mobilization to sustain financial viability of the university.

Program Objectives and its relationship to University Goals:

PROGRAM OBJECTIVES (PO)	**OBJECTIVES**						
A graduate of BS in Information Technology can:							
a. Innovate technological concepts and ideas underpinning desired IT solutions;	**a**	**b**	**c**	**d**	**e**	**f**	**g**
b. Administer competently the computer networks, system development, software applications, hardware and maintenance;	/	/	/	/	/	/	/
c. Design industry-based applications, infrastructures and technologies that will promote the advancement and development of the community;	/	/	/	/	/	/	/
d. Adopt to various national and international industries standards in the practice of the profession; and;	/	/	/	/	/	/	/
e. Demonstrate professionalism in the social, environmental and legal aspects of Information Technology.	/	/	/	/	/	/	/

1. **Course Code** : IT223
2. **Course Title** : Event-Driven Programming
3. **Prerequisite** :
4. **Credits** : 3 UNITS

5. **Course Descriptions**

   This course introduces the fundamentals of event-driven programming, focusing on creating interactive applications that respond to user inputs and event. Students will learn to handle event such as mouse clicks, keyboard inputs, and system triggers, using frameworks like JavaScript, Java and C#. The course covers event handling, GUI components, asynchronous programming, and real-time applications.

6. Course Learning Outcomes and Relationship to Program Educational Objectives

COURSE LEARNING OUTCOMES	PROGRAM OBJECTIVES				
At the end of the semester, the students can:	a	b	c	d	e
a. Understand the principles and key concepts of event-driven programming	/	/	/	/	/
b. Implement event listener and handlers to manage user interactions in applications	/	/	/	/	/
c. Develop interactive GUI applications using event-driven programming techniques	/	/	/	/	/

7. Course Content

Course Objectives, Topics, Time allotment	Desired Student Learning Outcomes	Outcomes-Based Assessment (OBA) Activities	Evidence of Outcomes	Course Objectives	Program Outcomes	Values Integration
Topic: SKSU VMGO, Classroom Policies, Course Overview, Course Requirements, Grading System (2 hours)						
1. Discuss the VMGO of the university, classroom policies, scope of the course, course requirements and grading system	1.1 Student can be aware of and appreciate of the university's VMGO, classroom policies, course overview, requirements and grading system.	Individual participation in class discussion and group presentation	Individual participation in class discussion and group presentation			Value of appreciation

1. Introduction to Event-Driven Programming (lec:6hrs)						
1.1 Definition and principles of event-driven programming 1.2 Comparison with procedural and object-oriented paradigms 1.3 Event loop and event queue concepts	1.1 Define event-driven programming and explain its core principles, including the roles of events, event handlers, and asynchronous execution. 1.2 Differentiate event-driven programming from procedural and object-oriented paradigms by comparing their structure, flow control, and typical use cases. 1.3 Describe the concepts of an event loop and event queue, and explain how they manage the flow of events in an event-driven system.	Discussion Activities Recitation	Recitation Quizzes Laboratory activity	a	a, d, e	Unity and team work Value of participation Communication Challenge Achievement

2. Events and Event Handlers (lec:6hrs)						
2.1 Types of events (e.g., mouse, keyboard, window, custom) 2.2 Writing event listeners/handlers 2.3 Event binding and unbinding	2.1 Identify various types of events such as mouse clicks, keyboard input, window actions, and custom-defined events. 2.2 Explain the role of event listeners and handlers in responding to different events in an event-driven program. 2.3 Write appropriate event listeners and handlers for handling specific user or system interactions. 2.4 Demonstrate how to bind events to interface elements and how to unbind them when necessary. 2.5 Apply best practices in	Discussion Review	Rubrics score cards of laboratory exercise output accomplished by the instructor	b, c, d		Unity and team work Value of participation Communication Challenge Achievement

	managing event binding and unbinding to ensure efficient and error-free execution.					
3. Graphical User Interface (GUI) Programming (lec:6hrs)						
3.1 GUI Components (buttons, labels, text fields, checkboxes, etc.) 3.2 Layout management and containers 3.3 Handling user input through events	3.1 Identify and describe common GUI components such as buttons, labels, text fields, and checkboxes. 3.2 Utilize layout managers and containers to organize GUI components effectively within an interface. 3.3 Implement user interfaces using various GUI components and layout strategies. 3.4 Integrate event handling with GUI components to create responsive and interactive applications.	Discussion Recitation	Recitation Quizzes Laboratory activity	b, c, d		Unity and team work Value of participation Communication Challenge Achievement

4. Using Event-Driven Frameworks (lec:6hrs)						
4.1 JavaScript in browser (DOM events) 4.2 JavaFX/Swing (Java), Windows Forms/WPF (C#), or any language-specific framework 4.3 Event propagation and bubbling	4.1 Understand how JavaScript interacts with the DOM to handle events in a web browser, including even types, and handlers 4.2 Explore and apply the concepts of event handling using framework-specific tools, such as JavaFX, Swing (Java), Windows Forms, or WPF (C#) 4.3 Explain the concepts of event propagation and bubbling in event-driven programming and how events travel through the DOM or GUI components 4.4 Demonstrate how to manage event propagation and bubbling to control event flow	Discussion Recitation	Recitation Quizzes Laboratory activity	b, c, d		Unity and team work Value of participation Communication Challenge Achievement

	in a system, such as stopping propagation or using event delegation.					
5. Asynchronous Programming (lec:3hrs)						
5.1 Callbacks and promises (in JavaScript) 5.2 Async/await syntax 5.3 Handling concurrent events and race conditions	5.1 Explain the use of callbacks and promises in managing asynchronous operations in JavaScript 5.2 Differentiate between callbacks, promises, and async/await in terms of readability, structure, and use cases. 5.3 Write JavaScript code using async/await syntax to handle asynchronous tasks more efficiently	Discussion Recitation	Recitation Quizzes Laboratory activity	b, c, d		Unity and team work Value of participation Communication Challenge Achievement
6. Timers and Scheduled Events (lec:6hrs)						

6.1 Setting up interval or timeout-based events 6.2 Creating animations or real-time applications	6.1 Set up and manage interval-based (setInterval) and timeout-based (setTimeout) events in programming environments like JavaScript. 6.2 Explain the differences between interval and timeout mechanisms and their appropriate use cases 6.3 Implement time behaviors such as delayed execution or repeated tasks in event-driven applications	Discussion Recitation	Quizzes Laboratory activity	b, c, d		Unity and team work Value of participation Communication Challenge Achievement

6.4 Setting up interval or timeout-based events 6.5 Creating animations or real-time applications	6.4 Create basic animations or real-time features (e.g., clocks, live updates) using timed events and event-driven logic 6.5 Apply performance consideration when designing applications that rely on frequent or real-time updates.	Discussion Recitation	Quizzes Laboratory activity	b, c, d		Unity and team work Value of participation Communication Challenge Achievement
7. Real-Time and Interactive Applications (lec:3hrs)						
7.1 Games (basic mechanics using key/mouse events) 7.2 Multimedia controls	7.1 Implement basic game mechanics using keyboard and mouse events to control gameplay elements. 7.2 Design simple interactive games that respond to user inputs in real time through event-driven programming	Discussion Recitation	Recitation Quizzes Laboratory activity	b, c, d		Unity and team work Value of participation Communication Challenge Achievement

	7.3 Control multimedia elements (e.g., play, pause, volume, seek) using events listeners and handlers. 7.4 Integrate multimedia controls into applications to enhance user interaction and experience					
8. State Management in Event-Driven Systems (lec:3hrs)						
8.1 Managing program state in response to user events 8.2 Creating animations or real-time applications	8.1 Manage and update application state dynamically in response to various user-generated events. 8.2 Implement logic to maintain consistent state across interactive components in an event-driven application 8.3 Create responsive animations or real-time features	Discussion Recitation	Recitation Quizzes Laboratory activity	b, c, d		Unity and team work Value of participation Communication Challenge Achievement

	that reflect changes in application state.					
9. Debugging and Testing Event-Driven Code (lec:3hrs)						
9.1 Common issues (e.g., race conditions, memory leaks) 9.2 Creating animations or real-time applications	9.1 Identify common issues in event-driven programming, such as race conditions and memory leaks. 9.2 Explain the causes and consequences of these issues in the context of asynchronous and real-time applications 9.3 Design and implement animations or real-time features while ensuring efficient resource management	Discussion Recitation	Recitation Quizzes Laboratory activity	b, c, d, e		Unity and team work Value of participation Communication Challenge Achievement
10. Project-Based Learning (lec:6hrs)						
10.1 Develop a complete GUI application	10.1 Design and develop a fully functional GUI application using	Discussion Recitation	Recitation Quizzes Laboratory activity	b, c, d, e		Unity and team work Value of

10.2 Apply multiple event types and state management 10.3 Emphasize usability and responsiveness	event-driven programming principles 10.2 Integrate multiple event types 10.3 Manage application state effectively to ensure consistency and responsiveness in user experience 10.4 Apply design practices that enhance the usability and accessibility of the application					participation Communication Challenge Achievement
Examination (4 hours) Lectures (48 hours) **Total No. of Hours: 48 hours**						

7. Course Evaluation

Course Requirement: Students should have basic programming knowledge and be familiar with at least one language (JavaScript, Java, C#, or Python).

Grading System:

MIDTERM TERM

Exam 40%
Attendance 10%
Assignment/Quizzes 15%
Laboratory Exercise/Project 35%

FINAL TERM

Exam 40%
Attendance 10%
Assignment/Quizzes 15%
Laboratory Exercise/Project 35%

MTG+FTG/2=FG

Schedule of Examination

Midterm	- March 25-28, 2025
Final Term	- May 20-23, 2025

References:

TextBooks:

1. Sebesta, R. W. (2014). *Programming the World Wide Web* (8th ed.). Pearson.
2. Flanagan, D. (2020). *JavaScript: The Definitive Guide* (7th ed.). O’Reilly Media.
3. Horstmann, C. S. (2018). *Core Java Volume I – Fundamentals* (11th ed.). Pearson.
4. Balagurusamy, E. (2018). *Programming in C#: A Primer* (5th ed.). McGraw-Hill Education.

5. Moore, A. D. (2019). *Python GUI Programming with Tkinter: Develop responsive and powerful GUI applications with Tkinter*. Packt Publishing.
6. Sierra, K., & Bates, B. (2005). *Head First Java* (2nd ed.). O'Reilly Media
7. Horstmann, C. S. (2020). *Modern JavaScript for the Impatient*. Addison-Wesley Professional.
8. Perkins, B., Reid, J., & Jacob, T. (2018). *Beginning Visual C# and .NET* (2018 ed.). Wrox/Wiley.

Supplemental:

1. https://www.w3schools.com
2. https://docs.oracle.com/javase/tutorial
3. https://www.geeksforgeeks.org
4. https://www.khanacademy.org/computing/computer-programming/programming
5. https://stackoverflow.com
6. https://realpython.com
7. https://www.tutorialspoint.com

Prepared by:

KRISTINE MAE H. AMPAS, MIT
Faculty

Reviewed by:

CERILO B. RUBIN, JR, MIT
Faculty Program Chairperson, BSIT

Approved:

ELBREN O. ANTONIO, DIT
Dean, College of Computer Studies