Software Development Student Learning Outcomes

Knowledge		Student Learning Outcomes
K-24	Knowledge of continuous automation and production deployment practices.	Describe the tools and techniques used for automation in the integration and deployment of software systems.
K-22	Knowledge of AI and ML methods and algorithms.	Compare and Contrast Artificial Intelligence (AI) and Machine Learning (ML) principles and techniques.
K-38	Knowledge of the difference between AI and ML.	
K-39	Awareness of current and specialized AI and ML tools and their application to business problems.	Recognize current and specialized AI and ML tools to solve business problems.
K-17	Knowledge of common program architectures (e.g. standalone, three-tier, web-based, cloud-based, serverless, microservice).	Describe common application architectures.
K-16	Knowledge of block chain processes and practices.	Explain blockchain and its underlying technologies and practices.
K-35	Knowledge of implementation and utilization of cloud services including deployment (e.g. AWS, Microsoft Azure).	Describe tools, techniques, and frameworks provided by cloud vendors to support software development.
K-36	Knowledge of implementing edge-cloud software controls and services.	Explain how edge computing strategies can be applied to enhance the use of cloud services.
K-5	Knowledge of web services (e.g. service-oriented architecture, Simple Object Access Protocol, and web service description language).	List common services, practices, and protocols used in the development of web services, including the use of cookies.
K-11	Knowledge of the appropriate use of cookies.	
K-25	Knowledge of cybersecurity and privacy principles and methods that apply to software development.	Explain information security principles and fundamentals which apply to software development.
K-30	Knowledge of cyber threats and vulnerabilities.	Identify how to assess internal and external network systems threats and vulnerabilities.
K-33	Knowledge of basic security practices including threats and vulnerabilities that may arise from interactions with other systems, external and legacy code.	
K-32	Awareness of standards such as PCI, PHI, and GDPR.	Describe laws, regulations and standards related to cybersecurity and privacy globally.
K-27	Knowledge of code security (e.g. hashing, encryption, cryptography, threat modeling).	Explain secure coding algorithms such as hashing, encryption and cryptography.
K-40	Conceptual knowledge of PKI.	
K-37	Knowledge of software development and implementation for communicating and gathering data from IoT devices.	Describe the process of gathering data from IoT devices connectivity for software development and implementation.
K-42	Knowledge of structure and unstructured data sources.	Describe how to access data from different types of data sources.
K-21	Knowledge of database integration/management software.	Discuss database integration tools and techniques for software management.
K-3	Knowledge of effective software debugging principles.	Describe tools and techniques used to debug software applications.
K-19	Knowledge of server software patterns, messaging patterns both async and synch.	Describe common software design patterns and the characteristics of the applications which implement them.
K-45	Knowledge of best practices for Design/UI/UX/accessibility as applied to software development.	Define UI, UX, and accessibility and describe best practices for incorporating them in a software system design.
K-15	Knowledge of the differences between client-side scripting and server-side scripting.	Describe the differences between client-side and server-side scripting in software development.
K-49	Knowledge of process flow and how the upgrade/implementation of software is accomplished through definitive understanding of team collaboration in DevOps, End of Life Cycle, and including importance of foundational security.	Describe DevSecOps as it relates to DevOps and the secure software development life cycle.
K-6	Knowledge of UML documents that model a program.	Describe tools and techniques used for modeling and documenting software systems.
K-8	Knowledge of Software Integration Management Systems – how industry documents final product builds to show all of the elements that have changes and checks those that have not changed.	
K-20	Knowledge of Enterprise application integration software.	Discuss tools, techniques, and frameworks for integrating the components of an Enterprise application.
K-44	Knowledge of ethics and its application to software development.	Describe ethical concerns in software development, authoritative references for ethical standards in software development, and how decisions can be made using ethical reasoning.
K-7	Knowledge of how programs communicate across the Internet using conventions such as Remote Method Invocation.	Describe protocols used for software which operates across a network.
K-14	Knowledge of error handling constructs.	Describe how constructs are used to handle errors in software.
K-4	Knowledge of computer programming languages and principles in general.	List commonly used programming languages and general concepts that are common to these languages.
K-12	Knowledge of how applets differ from applications in terms of program form, operating context, and how they are started.	Explain the differences between applets and applications.
K-47	Knowledge of mobile application development.	Describe tools, techniques, and frameworks used for mobile application development.
K-34	Knowledge of computer network fundamentals (e.g. TCP/IP, HTTPS, ports, firewall, LAN/WAN etc.)and network security methodologies.	Describe network concepts which apply to software design and implementation.
K-13	Knowledge of two or more operating systems that are current industry standards (e.g. Linux, Windows Apple OS).	Describe common operating systems used for software applications.
K-29	Knowledge of risk management framework and processes (e.g. methods for assessing and mitigating risk).	Discuss risk management framework and processes.

K-1	Knowledge of software development models (e.g. Waterfall Model, Spiral Model).	Describe various software development models.
K-2	Knowledge of system design tools, methods, and techniques, including automated systems analysis and design tools.	Describe tools, methods, and techniques used for software analysis and design.
K-18	Knowledge of the local development cycle (e.g. build, deploy, test, debug).	List the phases of the software development lifecycle.
K-46	Knowledge of lifecycle development/steady state/end of life.	
K-10	Knowledge of Regression Testing Development – how to test software using software.	Describe automation techniques that can be applied to software testing.
K-9	Knowledge of event handling in a GUI.	Describe how event-handling is implemented in a GUI (graphical user interface).
K-26	Knowledge of system and application security threats and vulnerabilities (e.g. buffer overflow, mobile code, cross-site scripting, Procedural Language/Structured Query Language [PL/SQL] and injections, race conditions, covert channel, replay, return-oriented attacks, malicious code).	Identify and describe common application security vulnerabilities.
K-31	Knowledge of software related information technology (IT) security principles and methods (e.g. modularization, layering, abstraction, data hiding, simplicity/minimization).	Classify the application of secure coding principles and methods.
K-41	Knowledge of DevSecOps.	Describe DevSecOps principles and practices.
K-48	Knowledge of how to protect data privacy through code.	Explain how to protect data privacy by secure coding.
K-28	Knowledge of Privacy Impact Assessments in terms of privacy and identify management.	Describe how Privacy Impact Assessment (PIA) tools are used to identify and mitigate privacy risks.
K-23	Knowledge of software collaboration tools (e.g. version control, bug tracking, continuous integration).	Describe how software tools are used to collaborate and manage the phases of the software development lifecycle.
K-43	Knowledge of open source software.	Compare and contrast common open source frameworks and tools used for software development.
Skills		**Student Learning Outcomes**
S-1	Skill in using built-in functions as well as skill in creating custom functions, subroutines, and procedures within software using scripting languages.	Develop a modularized, distributed application using a scripting language.
S-15	Skill in using a scripting language on the server side and the client side of a distributed program.	
S-2	Skill in integrating standard object model components with server pages.	Develop a server page containing integrated object model components.
S-3	Skill in conducting software debugging.	Diagnose and resolve software defects using debugging tools and techniques.
S-4	Skill in creating programs that validate and process multiple inputs including command line arguments, environmental variables, and input streams.	Design and implement applications which acquire and validate input from various sources, including command line arguments, files, environment variables, and input streams.
S-5	Skill in writing code in a currently supported programming language (e.g. scripting ,Java, C++, Linux, Ruby or current languages).	Develop applications which uses a standard current programming language.
S-6	Skill in developing applications that can log and handle errors, exceptions, and application faults and logging.	Design resilient applications which provide comprehensive diagnostic information.
S-7	Skill in applying root cause analysis (RCA) techniques to solving software/customer issues.	Apply root cause analysis techniques to identify and diagnose software defects.
S-8	Skill in the live production environment (e.g. monitoring, logging, alerting, remote debugging).	Demonstrate familiarity and ability to work with common tools for monitoring and diagnosing issues in production systems.
S-9	Skill in using electronic mail software (e.g. Google Gmail; IBM Notes Hot technology; Microsoft Exchange Server Hot technology; Microsoft Outlook Hot technology).	Demonstrate the use of different electronic mail systems.
S-10	Skill in using graphical user interface development software (e.g. Graphical user interface GUI builder software; Graphical user interface GUI design software; Salesforce Visualforce Hot technology).	Use Graphical Use Interface (GUI) development tools to develop applications.
S-11	Skill in using object or component oriented development software (e.g. C++ Hot technology; Document Object Model DOM Scripting; Python Hot technology; Simple API for XML SAX).	Use component-driven development software tool to build software components.
S-12	Skill in creating classes that use inheritance aspects of the object-oriented paradigm.	Demonstrate the use of object-oriented features such as inheritance and polymorphism.
S-13	Skill in using, incorporating and utilizing cookies.	Develop web applications which use common services, practices, protocols, and cookies.
S-14	Skill in implementing programs that use local or remote databases with standard protocols.	Implement programs that use databases with standard protocols.
S-16	Skill in evaluating and reporting software needs, constraints, analysis for application-specific concerns.	Evaluate domain-specific needs, identify requirements, and define scope for applications.
S-17	Skill in implementing levels of security in distributed software applications and applets.	Apply secure development best practices and use standard tools to securely deploy applications.
S-18	Skill in deploying secure software according to secure software deployment methodologies, tools, and practices.	
S-19	Skill in designing software applications that are accessible by a variety of wireless and wired devices.	Develop applications which provide services for wired and wireless devices.
S-20	Skills such as time management, risk management.	Appropriately use time and risk management skills.
S-21	Skill in incorporating user experience feedback into software.	Apply iterative design techniques to the development of applications.
S-22	Skill in integrating third party open source resources into software.	Integrate third-party open source components into applications.

S-23	Skill in learning new and/or industry standard tools involved in the development of software.	Evaluate and learn current and new software tools and emerging tools as they become available.
Abilities		**Student Learning Outcomes**
A-1	Ability to both mentor and be mentored, provide critical feedback as well as accept critical feedback.	Perform the responsibilities of being a mentor and a mentee.
A-2	Ability to comprehend and execute both written and oral instructions by asking clarifying questions.	Demonstrate effective technical communication skills (both oral and written) to stakeholders.
A-3	Ability to effectively communicate technical concepts and constraints in written and oral form to technical team members, stakeholders.	
A-4	Ability to work effectively in multi-disciplinary teams to apply information technology in support of organizational goals.	Operate proficiently in multi-disciplinary teams to support organizational information technology goals.
A-9	Ability to engage with users and understand their user experience.	
A-5	Ability to write technical documentation for technical and nontechnical audiences.	Construct technical documents for appropriate audience.
A-6	Ability to manage your own software development project activities and deliverables in a timely and efficient manner.	Organize and schedule software projects to meet one's own deliverables timeline.
A-7	Ability to work on team projects and demonstrate critical thinking, teamwork, oral communications, inter-cultural appreciation, and technical and information literacy skills.	Demonstrate effective team collaboration and communications skills for technical and information proficiency.
A-8	Ability to research and be able to find other sources to answer the problem.	Research and utilize validated data to solve the given problem.
A-10	Ability to draw on prior knowledge and experience in a new situation.	Make use of prior knowledge and experience in new situations.