ILLINOIS TECH | College of Computing

Department of Information Technology and Management *Course Rubric*

ITM 313 RUBRIC

ITM 313 Introduction to Open Source Application Development
Students may be scored on a scale of 1 to 5; scores of 2 and 4 may be interpolated.

Program Educational Objectives			
Objective Score ▶	**5**	**3**	**1**
Perform requirements analysis, design and administration of computer and network-based systems conforming to policy and best practices, and monitor and support continuing development of relevant policy and best practices as appropriate.	The student is consistently able to perform requirements analysis, to design and administer computer and network-based systems conforming to policy and best practices, and to monitor and support continuing development of relevant policy and best practices as appropriate	The student is generally able to perform requirements analysis, to design and administer computer and network-based systems conforming to policy and best practices, and to monitor and support continuing development of relevant policy and best practices as appropriate, but this may not be consistent	The student is unable to perform requirements analysis, to design and administer computer and network-based systems conforming to policy and best practices, or to monitor and support continuing development of relevant policy and best practices
Course student outcomes Upon completion of this course the student should be able to do the following:			
Outcome Score ▶	**5**	**3**	**1**
Recall and describe software application and development theory and concepts	The student is able to recall and describe software application and development theory and concepts accurately and in detail	The student is able to recall and describe software application and development theory and concepts with occasional omissions	The student is unable to recall and describe software application and development theory and concepts
Write, compile, execute, troubleshoot, analyze, evaluate, and resolve simple problems through program coding using the Python programming language	The student is consistently able to write, compile, execute, troubleshoot, analyze, evaluate, and resolve simple problems through program coding using the Python programming language	The student is able with some assistance to write, compile, execute, troubleshoot, analyze, evaluate, and resolve simple problems through program coding using the Python programming language	The student is unable to write, compile, execute, troubleshoot, analyze, evaluate, and resolve simple problems through program coding using the Python programming language
Develop, synthesize, and identify important language-standard libraries and utilities	The student is consistently able to develop, synthesize, and identify important language-standard libraries and utilities	The student is basically able to develop, synthesize, and identify important language-standard libraries and utilities, but may require some assistance	The student is not able to develop, synthesize, and identify important language-standard libraries and utilities
Apply data transfer techniques between modules using parameters and return values	The student is consistently able to apply data transfer techniques between modules using parameters and return values	The student is generally able to apply data transfer techniques between modules using parameters and return values, but may require some assistance	The student is unable to apply data transfer techniques between modules using parameters and return values
Construct applications to use simple files for input and output	The student is consistently able to construct applications to use simple files for input and output	The student is generally able to construct applications to use simple files for input and output, but may require some assistance	The student is unable to construct applications to use simple files for input and output
Implement arrays as structures to contain data	The student is able to implement arrays as structures to contain data	The student is generally able to implement arrays as structures to contain data, but may require some assistance	The student is unable to implement arrays as structures to contain data
Use a higher-level programming language to code, test, and debug software designs	The student is consistently able to use a higher-level programming language to code, test, and debug software designs	The student is generally able to use a higher-level programming language to code, test, and debug software designs, but may require some assistance	The student is unable to use a higher-level programming language to code, test, and debug software designs
Implement concepts of Object Oriented Programming (OOP), inheritance and polymorphism	The student is fully able to implement concepts of Object Oriented Programming (OOP), inheritance and polymorphism	The student is frequently able to implement concepts of Object Oriented Programming (OOP), inheritance and polymorphism	The student is unable to implement concepts of Object Oriented Programming (OOP), inheritance and polymorphism
Describe integration of Graphical User Interfaces (GUIs) and event driven programming	The student is able to describe integration of Graphical User (GUIs) and event driven programming accurately and in detail	The student is able to describe integration of Graphical User Interfaces (GUIs) and event driven programming with some omissions or inaccuracies	The student is unable to describe integration of Graphical User Interfaces (GUIs) and event driven programming
Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	The student is consistently able to analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	The student is, under most circumstances, able to analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	The student is unable to analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions
Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	The student is consistently able and prepared to design, implement, and evaluate a computing-based solution to meet a given set of computing requirements	The student in most cases is able and prepared to design, implement, and evaluate a computing-based solution to meet a given set of computing requirements	The student is not able to design, implement, and evaluate a computing-based solution to meet a given set of computing requirements
Identify and analyze user needs and take them into account in the selection, creation, evaluation, and administration of computer-based systems	The student is always able to identify and analyze user needs and take them into account in the selection, creation, evaluation, and administration of computer-based systems	The student is occasionally able to identify and analyze user needs and take them into account in the selection, creation, evaluation, and administration of computer-based systems, but not necessarily consistently	The student is unable to identify and analyze user needs and take them into account in the selection, creation, evaluation, and administration of computer-based systems