**DESCRIPTOR: ITIS 130**

|  |  |  |  |
| --- | --- | --- | --- |
| Discipline: Information Technology and Information Systems | Proposed Sub-discipline (if applicable): | | |
| General Course Title:  **Introduction to Programming Concepts and Methodologies** | | | Min. Units 3 |
| General Course Description:  An introduction to the fundamental concepts and models of application development including the basic concepts of program design, data structures, programming, problem solving, programming logic, and fundamental design techniques for event-driven programs. Hands-on experience with a modern application programming language and development platform. | | | |
| Proposed Number: ITIS 130 | Proposed Suffix (if applicable): | | |
| Required Prerequisites: None | | | |
| Required Co-Requisites[[1]](#footnote-1): None | | | |
| Advisories/Recommended Preparation[[2]](#footnote-2):  The course benefits from computer lab resources either in class or available for licensing on individual students’ computers. The choice of language should reflect commonly used languages and tools with the expectation that learning any language will generalize to other languages. For this reason it may be best to concentrate on one language to develop depth rather than breadth across several languages. | | | |
| Course Content:   1. Program design 2. Program development lifecycle 3. Requirements determinants and analysis 4. Modular design 5. Techniques for modeling program structures 6. Programming concepts 7. Variables 8. Literals 9. Types 10. Expressions 11. Procedures 12. Functions 13. Parameters 14. Operators and operations 15. Decision logic 16. Looping 17. Sub-procedures 18. Passing parameters 19. Coding 20. Unit testing 21. Control structures   Some the above material is taken from <http://www.acm.org/education/curricula/IS%202010%20ACM%20final.pdf>. | | | |
| Course Objectives: *At the conclusion of this course, the student should be able to:*   1. use primitive data types and data structures offered by the development environment. 2. choose an appropriate data structure for modeling a simple problem. 3. identify basic programming concepts. 4. write simple applications that relate to a specific domain. 5. design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, and the definition of functions. 6. test applications with sample data. 7. apply core program control structures. | | | |
| Methods of Evaluation:  Evaluation will include hands-on projects and a combination of examinations, presentations, discussions, or problem-solving assignments. | | | |
| Sample Textbooks, Manuals, or Other Support Materials (do not include editions or publications dates):   * Liang, Y. - Introduction to Java Programming, Comprehensive Version * Deitel, P. & Deitel, H. - How to Program * Guttag, J. - Introduction to Computation and Programming Using Python: With Application to Understanding Data * Gaddis, T. & Irvine, K. - Starting Out With Visual Basic * Mansfield, R. - Mastering VBA for Microsoft Office * Murach, M., Prince, A. & Menendez, R. - Murach's Mainframe COBOL | | | |
| FDRG Lead Signature: Markus Geissler, PhD Date: 20Jan2021 | | | |
| [For Office Use Only] | | **Internal Tracking Number** | |
|  | | | |
|  | | | |

1. Prerequisite or co-requisite course need to be validated at the CCC level in accordance with Title 5 regulations; co-requisites for CCCs are the linked courses that must be taken at the same time as the primary or target course. [↑](#footnote-ref-1)
2. Advisories or recommended preparation will not require validation but are recommendations to be considered by the student prior to enrolling. [↑](#footnote-ref-2)