Information Technology Advisory Meeting

Fall 2015 Agenda

February 12, 2016 @ 1:00 pm

MSCTC Moorhead B142

**Agenda for spring 2016**

* MSCTC/M State welcome
* Introductions
* Review/Revision/Approval of last meetings minutes
* Membership list additions and/or removals
* WebEx procedure – virtual attendees
* Core Class/competency listing
  + M State is working on better coring our curriculum offering
  + This list is a suggested list of core classes/competencies to begin the discussion. The group is welcome to add/subtract from the list. Depending on the time and pleasure of the group, the faculty is interested the group’s opinion of core or non-core for each item. It is possible to detail core for selected majors.
    - Public Speaking
    - Cisco 1 or Network +
    - Database
    - Robotics – Robotics Programming
    - College Writing
    - Linux
    - Introduction to Programming & Scripting (Python 1)
    - Informatics
  + This list is non-core classes/competencies to begin the discussion. The group is welcome to add/subtract from the list.
    - Information Systems
    - Microcomputer Maintenance
    - Web Engineering 1
    - Microcomputer Operating System – Windows client
    - General Psychology
    - College Algebra
    - Network Security
    - Network Operating System (Server Administration)
    - Cisco 2
    - Managing Directory Services
    - Web Security
    - Security Breaches & Countermeasures
    - VoIP
    - Internship or Supervised Occupational Experience
    - Network Defense
    - Enterprise Network Technologies
    - SQL
    - Visual Basic Programming 1
    - Java Programming
    - COBOL Programming
    - Computer Science 1
    - RPG Programming
    - Mobile Application Development
    - Database Integration
  + This is the two year IT Competency Model. Where would the group like to see these items, core or non-core? (<http://ccecc.acm.org/files/publications/ACMITCompetencyModel14October201420150114T180322.pdf>
    - An ability to demonstrate core IT competency in client computing and user support
      * Carry out trouble-shooting strategies for resolving an identified end-user IT problem.
      * Differentiate among various operating systems.
      * Explain the process of authentication and authorization between end-user devices and computing network resources.
      * Identify a variety of assistive or adaptive technologies and universal design considerations.
      * Identify basic components of an end-user IT system.
      * Implement a hardware and software configuration responsive to an identified scenario.
      * Summarize life-cycle strategies for replacement, reuse, recycling IT technology and resources.
      * Summarize strategies to support or train users with their IT resources.
      * Use a variety of practices for making end-user IT systems secure.
    - An ability to demonstrate core IT competency in database and information management
      * Describe the data management activities associated with the data lifecycle.
      * Diagram a database design based on an identified scenario.
      * Differentiate between public and private data.
      * Discuss applications of data analytics.
      * Discuss issues relevant to dealing with very large data sets, both structured and unstructured.
      * Identify database administration tasks.
      * Produce simple database queries.
      * Use data analytics to support decision making for a given scenario.
    - An ability to demonstrate core IT competency in digital media and immersive technology
      * Differentiate among a variety of technology-based sensory interactions.
      * Differentiate among data types, data transfer protocols and file characteristics specific to the targeted use.
      * Illustrate the activities of a digital media design process.
      * Implement communication principles into digital media design.
    - An ability to demonstrate core IT competency in networking and convergence
      * Carry out basic computer network troubleshooting techniques.
      * Describe the layers, protocols and components of the OSI model.
      * Diagram the components of an integrated IT system.
      * Differentiate among various computer networking models.
      * Differentiate among various techniques for making a computer network secure.
      * Summarize the flow of data through a computer network scenario.
    - An ability to demonstrate core IT competency in programming and application development
      * Demonstrate best practices for designing end-user computing interfaces.
      * Demonstrate the techniques of defensive programming and secure coding.
      * Diagram the phases of the Secure Software Development Lifecycle.
      * Discuss software development methodologies.
      * Summarize the differences among various programming languages.
      * Use a programming or a scripting language to share data across an integrated IT system.
      * Use a programming or a scripting language to solve a problem.
    - An ability to demonstrate core IT competency in servers, storage and virtualization
      * Differentiate among strategies for business continuity provisioning of IT resources at the enterprise level.
      * Discuss data governance and its implications for users as well as IT professionals.
      * Identify a variety of enterprise-level digital storage technologies.
      * Implement an application of virtualization.
      * Modify a system to improve data confidentiality or regulatory compliance.
      * Summarize the implications of various cloud computing models.
      * Summarize the security implications and risks for distributed IT systems.
    - An ability to function effectively as a member of a diverse team to accomplish common goals
      * Use communication, negotiation, and collaboration skills as a member of a diverse team.
    - An ability to read and interpret technical information, as well as listen effectively to, communicate orally with, and write clearly for a wide range of audiences
      * Describe the attitudes, knowledge and abilities associated with quality customer service.
      * Produce technical documentation responsive to an identified computing scenario.
      * Use documentation or a knowledge base to resolve a technical challenge in an identified computing scenario
    - An ability to engage in continuous learning as well as research and assess new ideas and information to provide the capabilities for lifelong learning
      * Discuss significant trends and emerging technologies and their impact on our global society.
    - An ability to exhibit professional, legal, and ethical behavior
      * Demonstrate professional behavior in response to an ethically-challenging scenario in computing.
      * Summarize the tenets of ethics and professional behavior promoted by international computing societies.
    - An ability to demonstrate business awareness and workplace effectiveness
      * Describe IT procurement processes for goods and services.
      * Summarize the role of IT in supporting the mission and goals of an organization.
  + This is the two year Computer Engineering Competency Model. Where would the group like to see these items, core or non-core? (<http://ccecc.acm.org/guidance/software-engineering/objectives/>
    - CS. 4. Analyze the efficiency of recursive algorithms.
    - CS. 5. Analyze the execution of searching and sorting algorithms.
    - CS. 6. Apply fundamental theorems and rules of Calculus to differentiate and integrate algebraic, trigonometric, inverse trigonometric and transcendental functions.
    - CS. 7. Apply fundamental theorems and rules of Calculus to evaluate limits and analyze the continuity of various functions.
    - CS. 8. Apply fundamental theorems and rules of differentiation to solve problems that model real-world situations.
    - SE. 1. Apply mathematical induction and other techniques to prove mathematical results.
    - CS. 9. Apply secure coding techniques to object‐oriented programming solutions.
    - CS. 10. Apply the program development process to problems that are solved using fundamental programming constructs and predefined data structures.
    - CS. 11. Assess the appropriateness of using recursion to solve a given problem.
    - CS. 12. Compare and contrast a range of searching and sorting algorithms and analyze time and space efficiencies.
    - CS. 13. Compare and contrast the primitive data types of a programming language; describe how each is stored in memory; and identify the criteria for selection.
    - SE. 2. Construct a preliminary investigative report for a proposed system that includes scheduling and plans for mitigating potential risks.
    - CS. 14. Construct multiple‐file or multiple‐module programming solutions that use class hierarchies, inheritance, and polymorphism to reuse existing design and code.
    - CS. 15. Construct object oriented programming solutions for reuse, using ADTs that incorporate encapsulation, data abstraction, and information hiding.
    - CS. 16. Construct symbolic models of applied problems described in words.
    - CS. 17. Create effective, efficient and secure software, reflecting standard principles of software engineering and software assurance.
    - CS. 18. Create programming solutions that use data structures and existing libraries.
    - CS. 19. Decompose a program into subtasks and use parameter passing to exchange information between the subparts.
    - SE. 3. Decompose complex systems using best practice object-oriented analysis and design tools and techniques.
    - CS. 20. Describe the language translation phases of compiling, interpreting, linking and executing, and differentiate the error conditions associated with each phase.
    - CS. 21. Design and construct programming solutions using a variety of recursive techniques.
    - CS. 22. Design and develop reusable software using appropriate data structures and templates.
    - CS. 23. Design and develop secure and fault-tolerant programs that mitigate potential security vulnerabilities.
    - SE. 7. Design and implement consistent and coherent user-centered interfaces that comply with UI standard practices.
    - CS. 24. Differentiate between the object-oriented, structured, and functional programming methodologies.
    - SE. 4. Evaluate and test software system requirements that considers both validation and verification.
    - CS. 25. Examine the logical validity of arguments and proofs as they apply to Boolean expressions.
    - CS. 26. Illustrate the basic terminology and properties of graphs and trees.
    - CS. 27. Perform binary and hexadecimal conversions of numbers.
    - CS. 28. Perform computations using recursively defined functions and structures.  
      CS. 29. Produce algorithms for solving simple problems and trace the execution of computer programs.
    - CS. 30 Produce graphical user interfaces that incorporate simple color models and handle events.
    - SE. 5. Solve problems involving sets, relations, functions, and congruences.
    - CS. 31. Use fundamental concepts of Calculus to construct graphs of polynomial, rational and exponential functions.
    - CS. 32. Use graphs and trees to solve problems algorithmically.
    - SE. 6. Use methods of combinatorics to solve counting problems.
    - CS. 33. Use standard analysis and design techniques to produce a team-developed, medium-sized, secure software application that is fully implemented and formally tested.
    - CS. 34. Verify program correctness through the development of sound test plans and the implementation of comprehensive test cases.
    - An ability to exhibit professional, legal, and ethical behavior.
    - An ability to read and interpret technical information, as well as listen effectively to, communicate orally with, and write clearly for a wide range of audiences.
    - An ability to apply knowledge of computing and mathematics appropriate to the discipline.
    - An ability to interpret data, think critically and apply the scientific method.
    - An ability to demonstrate social awareness, respect for privacy and responsible conduct.
    - An ability to analyze a problem and craft an appropriate algorithmic and/or engineering solution.
    - An ability to design a secure system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.
    - An ability to analyze the global impact of software solutions on individuals, organizations, and society.
    - An ability to function effectively as a member of a multidisciplinary team to accomplish common goals.
* Discussion: Identify current and future trends/technologies
  + Topics, concerns and ideas committee members wish to communicate to MSCTC/MState
* Curriculum Updates
  + Computer Programming – AAS
    - Current curriculum review/discussion
    - Committee Recommendations
  + Network Administration and Security – AAS
    - Current curriculum review/discussion
    - Committee Recommendations
  + Information Technology – AS
    - Current curriculum review/discussion
    - Current articulation agreement expires May 30, 2018
    - ACM SIGITE guidelines on <http://brazil.minnesota.edu/advisory/>
    - Committee Recommendations

Advisory Member Functions (MSCTC Advisory Committee Guide)

* Identify specific subject areas of program inclusion
* Prioritizing the recommend subject areas
* Specifying appropriate program content level
* Reviewing program outcomes on an ongoing basis
* Assessment of program quality
* Specifying appropriate foundational skill standards for local needs
* Identifying general education and related technical skills needed by graduates
* Recommending equipment to support the program content