SYLLABUS

DATE OF LAST REVIEW: 02/2013
CIP CODE: 15.1201
SEMESTER: Departmental Syllabus
COURSE TITLE: Software and Hardware Concepts
COURSE NUMBER: CIST-0184
CREDIT HOURS: 3
INSTRUCTOR: Departmental Syllabus
OFFICE LOCATION: Departmental Syllabus
OFFICE HOURS: Departmental Syllabus
TELEPHONE: 913-334-1100
EMAIL: KCKCC issued email accounts are the official means for electronically communicating with our students.

PREREQUISITE(S): CIST-0101 Computer Concepts and Applications or CIST-0111 Microcomputer Business Software

REQUIRED TEXT AND MATERIALS: Please check with the KCKCC bookstore, http://www.kckcecbookstore.com/, for the required texts for your particular class.

COURSE DESCRIPTION: Software and Hardware Concepts is a technical survey of computer systems. This course emphasizes the interrelationships of hardware architecture, system software, and applications software. Topics of discussion include computer architecture, processors and storage systems, implications for system software design, and the effects of hardware architecture and system software on the development of applications programs in a business environment.

METHOD OF INSTRUCTION: A variety of instructional methods may be used depending on content area. These include but are not limited to: lecture, multimedia, cooperative/collaborative learning, labs and demonstrations, projects and presentations, speeches, debates, and panels, conferencing, performances, and learning experiences outside the classroom. Methodology will be selected to best meet student needs.
COURSE OUTLINE:
I. Algorithms.
   A. The attributes of algorithms, choose the appropriate algorithm, and measure the efficiency of algorithms.
II. Binary numbers, Boolean logic, and gates.
   A. Numbers in binary form.
   B. Gates and truth tables using Boolean logic.
   C. The basic circuit.
III. The von Neumann architecture.
   A. The simple von Neumann machine.
IV. Virtual machine perspective.
   A. The virtual machine.
   B. The types of system software.
   C. The assembly language.
V. High-level languages.
   A. Choosing the appropriate language for a project.
   B. The basic C++ program.
   C. Input/output statements, assignment statements, and control statements.
   D. The software life cycle.
VI. Procedural languages, object-oriented languages, and special purpose languages.
   A. Differentiating between the different languages and when they should be used.
VII. Alternative programming paradigms.
   A. Functional programming, logic programming, and parallel programming.
VIII. The compilation process.
   A. The compilation process and the four phases involved:
      1. lexical analysis,
      2. parsing, semantics
      3. code generation, and code optimization.
IX. The model of a computing agent.
   A. Turing machine simulator.
   B. Designing a Turing machine simulator.
X. Neural networks and artificial intelligence.
   A. Neural networks and recognition tasks with respect to artificial intelligence.
   B. Expert systems.
XI. Social and legal issues involved with the use of computer systems.
   A. Social and legal issues that relate to the use of computer systems:
      1. privacy, crime,
      2. constitutional and civil liberties,
      3. and encryption and wiretapping.

EXPECTED LEARNER OUTCOMES:
A. Upon completion of the course, the student will be able to define and design algorithms.
B. Upon completion of the course, the student will be able to manipulate binary numbers, understand Boolean logic, and identify gates.
C. Upon completion of the course, the student will have a firm understanding of the Von Neumann architecture.
D. Upon completion of the course, the student will be able to identify computers from a virtual machine perspective.
E. Upon completion of the course, the student will be able to identify high-level languages.
F. Upon completion of the course, the student will be able to identify procedural languages, object-oriented languages, and special purpose languages.
G. Upon completion of the course, the student will be able to identify alternative programming paradigms.
H. Upon completion of the course, the student will have a thorough understanding of the compilation process.
I. Upon completion of the course, the student will be able to identify a model of a computing agent.
J. Upon completion of the course, the student will be able to use neural networks and have a sound understanding of artificial intelligence.
K. Upon completion of the course, the student will be able to identify social and legal issues involved with the use of computer systems.

COURSE COMPETENCIES:

Upon completion of the course, the student will be able to define and design algorithms.
1. The student will be able to identify the attributes of algorithms, choose the appropriate algorithm, and measure the efficiency of algorithms.

Upon completion of the course, the student will be able to manipulate binary numbers, understand Boolean logic, and identify gates.
2. The student will be able to represent numbers in binary form.
3. The student will be able to identify gates and construct truth tables using Boolean logic.
4. The student will be able to design and construct a basic circuit.

Upon completion of the course, the student will have a firm understanding of the von Neumann architecture.
5. The student will complete lab work, which will provide a hands-on experience with a simple von Neummann machine.

Upon completion of the course, the student will be able to identify computers from a virtual machine perspective.
6. The student will be able to define a virtual machine.
7. The student will be able to identify types of system software.
8. The student will be able to give a definition for an assembly language and with practical applications.

Upon completion of the course, the student will be able to identify high-level languages.
9. The student will know how to choose the appropriate language for a project.
10. The student will be able to develop a basic C++ program.
11. The student will be able to construct input/output statements, assignment statements, and control statements.
12. The student will be able to identify the software life cycle.
Upon completion of the course, the student will be able to identify procedural languages, object-oriented languages, and special purpose languages.

13. The student will be able to differentiate between the different languages and know when they should be used.

Upon completion of the course, the student will be able to identify alternative programming paradigms.

14. The student will have a through understanding of functional programming, logic programming, and parallel programming.

Upon completion of the course, the student will have a thorough understanding of the compilation process.

15. The student will complete exercises designed to illustrate the compilation process and the four phases involved: lexical analysis, parsing, semantics and code generation, and code optimization.

Upon completion of the course, the student will be able to identify a model of a computing agent.

16. The student will complete lab exercises in which examples of a Turing machine simulator.

17. The student will design a Turing machine simulator.

Upon completion of the course, the student will be able to use neural networks and have a sound understanding of artificial intelligence.

18. The student will complete lab exercises to illustrate the use of neural networks and recognition tasks with respect to artificial intelligence.

19. The student will work with expert systems and have a firm understanding of their use.

Upon completion of the course, the student will be able to identify social and legal issues involved with the use of computer systems.

20. The student will complete research involving the following social and legal issues that relate to the use of computer systems: privacy, crime, constitutional and civil liberties, and encryption and wiretapping.

21. The student will be able to discuss the ethics of the proper usage of personal consumer information in held in computer databases.

ASSESSMENT OF LEARNER OUTCOMES:
Assessment methods may include, but are not limited to, the following: Homework, Assignments, Quizzes, Class Participation, Chapter Tests, and Final Exam. The grading scale and the process for calculating the course grades are to be determined by the individual instructors. This information will be included in each instructor’s syllabus.
SPECIAL NOTES:
This syllabus is subject to change at the discretion of the instructor. Material included is intended to provide an outline of the course and rules that the instructor will adhere to in evaluating the student’s progress. However, this syllabus is not intended to be a legal contract. Questions regarding the syllabus are welcome any time.

Kansas City Kansas Community College is committed to an appreciation of diversity with respect for the differences among the diverse groups comprising our students, faculty, and staff that is free of bigotry and discrimination. Kansas City Kansas Community College is committed to providing a multicultural education and environment that reflects and respects diversity and that seeks to increase understanding.

Kansas City Kansas Community College offers equal educational opportunity to all students as well as serving as an equal opportunity employer for all personnel. Various laws, including Title IX of the Educational Amendments of 1972, require the college’s policy on non-discrimination be administered without regard to race, color, age, sex, religion, national origin, physical handicap, or veteran status and that such policy be made known.

Kansas City Kansas Community College complies with the Americans with Disabilities Act. If you need accommodations due to a documented disability, please contact the Director of the Academic Resource Center at 913-288-7670.