RARITAN VALLEY COMMUNITY COLLEGE
ACADEMIC COURSE OUTLINE

CSIT 256 Computer Architecture & Assembly Language

I. Basic Course Information

A. Course Number and Title:

CSIT 256, Computer Architecture & Assembly Language

B. New or Modified Course: Modified

C. Date of Proposal: Semester: Fall Year: 2018

D. Effective Term: Fall 2019

E. Sponsoring Department: Mathematics & Computer Science

F. Semester Credit Hours: 4

G. Weekly Contact Hours: Lecture: 3
Laboratory: 2
Out of class student work per week: 7

H. Prerequisite: CSIT 254 Data Structures & MATH 151 Calculus I or equivalent.

I. Laboratory Fees: Yes, at current rate

J. Name and Telephone Number or E-Mail Address of Department Chair and Divisional Dean at time of approval: Lori Austin – lori.austin@raritanval.edu (Chair),
Sarah Imbriglio – Sarah.Imbriglio@raritanval.edu (Divisional Dean)

II. Catalog Description

Prerequisite: CSIT ***254 Data Structures & MATH 151 Calculus I or equivalent.*** This course is the third in the sequence for students in Computer Science planning to transfer to a four-year college. It may also be taken as a free elective by interested students with sufficient background. This course focuses on the components of a computer that describe its architecture: storage, the central processing unit, the instruction set and addressing modes. The course also examines the way these components are interconnected and the nature of the information flow between them. Students will use Assembly language to reinforce these concepts.

III. Statement of Course Need

A. A Computer Scientist needs to understand the underlying architecture of the processor and the components of the computer that the processor interacts with to write programs effectively even with a high level language.

   The Association for Computing Machinery (ACM) requires this course for all Computer Science Graduates. It meets the transfer requirement to four year colleges and universities.

B. This course has a weekly lab component. The lab is essential for providing students hands on programming to write Assembly programs to explore the architecture of the processor

C. This course generally transfers as a Computer Science program requirement

IV. Place of Course in College Curriculum

A. Free Elective
B. This course meets a program requirement for Computer Science AS
C. Computer Elective on the Computer and Programming Electives List
D. Programming Elective on the Computer and Programming Electives List
E. To see course transferability: a) for New Jersey schools, go to the NJ Transfer website, www.njtransfer.org; b) for all other colleges and universities, go to the individual websites.

V. Outline of Course Content

A. Computer Architecture:
   1. Overview of Computer Architecture
   2. Binary Numbers
   3. Digital Logic
   4. Interconnections (System Bus, Expansion Bus)
   5. Cache Memory
   6. Internal Memory
   7. External Memory
   8. Input/Output
   9. Instruction Set
   10. Reduced Instruction Set (RISC)
   11. Parallel Architectures
   12. Intel IA-64 Architecture

B. Assembly Language
   1. Assembly Language Concepts
   2. Intel IA-32 Architecture

3. Binary Numbers and Big vs. Little Endian Numbers
4. Assembly Fundamentals
5. Using the Assembler
6. Data Transfer
7. Memory Addressing on IA-32
8. Integer Arithmetic
9. Procedures
10. Conditional Processing
11. Strings and Arrays
12. Interrupts
13. Structures/Macros
14. Disk Storage and File Processing

VI. General Education and Course Learning Outcomes

A. General Education Learning Outcomes

At the conclusion of the course, students will be able to:

1. Apply creative and critical thought in designing computing solutions that demonstrate knowledge of the computer architecture (GE-NJ 2, GE-NJ 4)
2. Apply quantitative reasoning to interpret data used in solving problems (GE - NJ 2)

B. Course Learning Outcomes

At the conclusion of the course, students will be able to:

1. Describe the main components of computer systems that define its architecture (CPU, storage, memory, instruction sets, and addressing modes)
2. Discuss the way the main components of computers are interconnected
3. Recognize assembly language syntax while reading and analyzing assembly language programs
4. Design, develop and test programs using MS Assembly Language commands while featuring various basic Assembly Language operations (data/program transfer, arithmetic instructions, indirect memory, addressing, procedures and stack operations)
5. Design, develop and test programs in the MS Assembly Language that include strings, arrays, macros, and conditional processing (Boolean instructions, loops)

C. Assessment Instruments

1. Labs – In-Class assignments
2. Projects – In-class and out of class projects
3. Exam - Exams on Computer Architecture
4. Exam – Exams on Assembly Language (paper-based and hands on)
5. Other – Homework on Architecture and Assembly

VII. Grade Determinants

A. Labs
B. Projects
C. Exams
D. Homework

Methods for teaching and learning that may be used in the course:
A. Lecture – Lecture on Computer Architecture
B. Lecture/Discussion – Lecture on Assembly Language with demonstration of programming in assembly.
C. Laboratory – Lab time to analyze, design, and write Assembly Programs.

VIII. Textbook: Suggestions

Computer Organization and Architecture: Designing for Performance, William Stallings, Prentice Hall, Tenth Edition, 2015.

Assembly Language for x86 Processors, Kip R. Irvine, Prentice-Hall Inc. (Pearson Education), Seventh Edition, 2015

(Please Note: The course outline is intended only as a guide to course content and resources. Do not purchase textbooks based on this outline. The RVCC Bookstore is the sole resource for the most up-to-date information about textbooks.)

IX. Resources

A. Microsoft Windows
B. An Integrated Development Editor such as Visual Studio 2017 with C++ enabled which includes the Microsoft Assembler

X. Honors Option

n/a