

## **Syllabus**

**Course Number: CS 370**

**Course Title: Assembly Programming**

### **Course Description:**

Describes the elements and techniques of assembly language programming for microprocessors used in the IBM compatible family of microcomputers. Introduces computer architectures, and discusses the concepts of data representations, processing instructions, addressing modes, macros, functions and procedures and file I/O.

### **Prerequisite Courses:**

- CS208 Computer Science Fundamentals
- CS362 Data Structures

In order to successfully complete this course, students are expected to have taken the prerequisite CS362 course or its equivalent. From CS362, you should have a working knowledge of the following topics:

1. Practice with problem definition, solution construction and algorithmic development using top-down design techniques.
2. Ability to implement control structures used for sequencing, selection, and iteration.
3. Ability to choose and implement appropriate data structures including arrays, structures, linked lists, and file processing.
4. Understanding of modular code design and how to design test cases.

Although CS370 introduces Assembly Programming, it is also a Computer Science course that introduces advanced programming concepts. Consequently, the ability to program at the CS362 learning outcome level is essential for successful in this course.

### **Course Overview**

In this course you will learn about the architecture of a typical microprocessor and its role in a computer system. By studying the machine and assembly language of a particular processor, specifically the x86 family of processors, you will begin to understand how a high-level programming languages, such as C++ and Java, are actually translated (compiled), and prepared for execution, loaded, and executed on the computer.

You will become familiar with the basic tools used for low-level programming: the editor, assembler, linker, loader, debugger, and machine language monitor. Most importantly, you will understand how data is stored and manipulated at the lowest levels of machine activity, providing a solid framework for developing high-level programming skills.

### Course Outcomes:

Upon completion of this course, learners should be able to:

1. Describe how the fetch-execute cycle is used during runtime in a classical von Neumann machine.
2. Create programs using various instructional formats such as addresses per assembly language command, fixed versus variable length formats, and data addressing modes.
3. Compare and contrast Assembly language constructs with prior learning about high-level programming language (e.g., C++) constructs.
4. Create programs that perform subroutine calls using the Assembly programming language.
5. Explain how subroutine calls are handled at the assembly level.
6. Master how numeric and non-numeric data are represented in memory including converting numerical data from one base to another such as hexadecimal, binary, and decimal.
7. Design, develop, execute, and debug assembly language programs (including string manipulation, subroutine calls, and representing arrays).
8. Create and execute test plans to verify program code works correctly.

### Course Materials:

#### Required Texts:

Irvine, K. R. (2014). *Assembly language for x86 processors* (7th ed.). Upper Saddle River, NJ: Prentice Hall (Pearson Education). ISBN: 978-0133769401 .

#### Technology Tools:

Visual Studio 2013 Express containing Microsoft Assembler (MASM), which can be downloaded (free) from the textbook author's Website:

<http://www.kipirvine.com/asm/>

Select the "Getting started with MASM" link.

**Pre-Assignment:**

- Complete first assigned readings in the Course Assignments and Activities table below
- Be prepared to ask questions on unclear areas in the assigned reading
- Familiarize yourself with the procedures for accessing the MASM assembler (above)

**Online Format:** Sign on to [worldclass.regis.edu](http://worldclass.regis.edu) and become familiar with the course navigation of the Web Curriculum. Complete assignments above.

**Classroom-based Format:** Complete assignments above by the first night of class.



## Course Assignments and Activities:

	Topics	Readings	Activities Assignments and Associated Points*
1	<ul style="list-style-type: none"> <li>• Computer Number Systems and Architectures</li> <li>• Fundamental Assembly Language Elements: data definition and data movement instructions</li> </ul>	Textbook: Chapters 1, 2, 3, & Chapter 4, sec 4.1 Online Content for Week 1	Introductions Pretest Quiz Participation in Discussions (10% for all entire course) Programming Assn 1 (10%)
2	<ul style="list-style-type: none"> <li>• Basic Arithmetic and Loop Instructions</li> </ul>	Textbook: Chapter 4, sec 4.2 – 4.9 Online Content for Week 2	Participation in Discussions Programming Assn 2 (10%) Jesuit Key Values Essay (5%)
3	<ul style="list-style-type: none"> <li>• Procedures and the Stack and Conditional Processing Instructions</li> </ul>	Textbook: Chapter 5 Online Content for Week 3	Participation in Discussions Programming Assn 3 (10%)
4	<ul style="list-style-type: none"> <li>• Boolean Instructions and Implementing Control Structures in Assembly Language</li> </ul>	Textbook: Chapter 6 Online Content for Week 4	Participation in Discussions Programming Assn 4 (10%) Midterm (10%)
5	<ul style="list-style-type: none"> <li>• Shift and Rotate Instructions</li> </ul>	Textbook: Chapter 7, sec 7.1 – 7.3 Online Content for Week 5	Participation in Discussions Programming Assn 5 - Part I: Documentation
6	<ul style="list-style-type: none"> <li>• Parsing Integer Data, and Extended Precision Arithmetic</li> </ul>	Textbook: Chapter 7, sec 7.4 – 7.9 Online Content for Week 6	Participation in Discussions Programming Assn 5 - Part II: Program (10% for both parts)
7	<ul style="list-style-type: none"> <li>• Procedures and Parameters, Stack Frames, Multiple File Programs</li> </ul>	Textbook: Chapter 8 Online Content for Week 7	Participation in Discussions Programming Assn 6 - Part I: Documentation
8	<ul style="list-style-type: none"> <li>• String Instructions, String Representations and Array Representations and Processing</li> </ul>	Textbook: Chapter 9 Online Content for Week 8	Participation in Discussions Programming Assn 6 - Part II: Program (10% for both parts) Final exam (15%)
		<b>Total</b>	<b>100%</b>

**\*Note to Classroom sections only:** Exact due dates for programming assignments may be modified from what is indicated in the above **Course Assignments and Activities** grid. Your facilitator's syllabus, handed out the first night of class, will indicate any changes.

## Programming Assignments

Each programming assignment will involve designing, implementing, and testing a program using the concepts discussed in the book and class.

**Note:** programs that do not assemble, are not modular, nor properly documented standards will not be accepted.

Programming assignments will be graded according to details listed in the rubrics given in the class, which also includes the following criteria:

1. Source Code
  - a. Code is easy to read and self-documenting
  - b. Appropriate comments are used
2. Coding Style
  - a. When applicable, the program is modular
  - b. The Java programming language is used correctly
3. Functionality
  - a. There are no assembly, logic, and design errors.
  - b. The program solves the assigned problem.
4. Testing, as appropriate
5. Analysis (if required)
  - a. Logical and complete

## Exams

There will be a midterm and final exam. Exams questions will be cumulative, taken from reading assignments, programming assignments, and class participation. Exams may test your understanding, comprehension, application, and evaluation of the material presented in this class.

## Participation

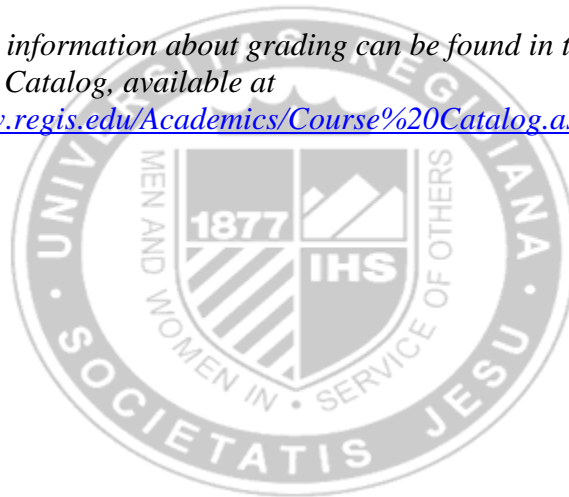
Class participation/effort is important because we can all learn from each other. Your participation points can make a difference in the final grade. Participation means:

1. Present in class every session (classroom)  
Present in the forum every week (online)
2. Effectively responds to questions from the facilitator (classroom)  
Regularly checks forum and posts all required items by the deadlines (online)
3. Interacts/replies to other students in classroom/forum discussions.

## CC&IS Grading Scale

Letter Grade	Percentage	Grade Point
A	93 to 100	4.00
A-	90 to less than 93	3.67
B+	87 to less than 90	3.33
B	83 to less than 87	3.00
B-	80 to less than 83	2.67
C+	77 to less than 80	2.33
C	73 to less than 77	2.00
C-	70 to less than 73	1.67
D+	67 to less than 70	1.33
D	63 to less than 67	1.00
D-	60 to less than 63	.67
F	Less than 60	0

*Additional information about grading can be found in the latest edition of the University Catalog, available at <http://www.regis.edu/Academics/Course%20Catalog.aspx>.*



## CC&IS Policies and Procedures

Each of the following CC&IS Policies & Procedures is incorporated here by reference. Students are expected to review this information each term, and agree to the policies and procedures as identified here and specified in the latest edition of the University Catalog, available at <http://www.regis.edu/Academics/Course%20Catalog.aspx> or at the link provided.

- The CC&IS Academic Integrity Policy.
- The Student Honor Code and Student Standards of Conduct.
- Incomplete Grade Policy, Pass / No Pass Grades, Grade Reports.
- The Information Privacy policy and FERPA. For more information regarding FERPA, visit the [U.S. Department of Education](http://www.ed.gov).
- The HIPAA policies for protected health information. The complete Regis University HIPAA Privacy & Security policy can be found here: <http://www.regis.edu/About-Regis-University/University-Offices-and-Services/Auxiliary-Business/HIPAA.aspx>.
- The Human Subjects Institutional Review Board (IRB) procedures. More information about the IRB and its processes can be found here: <http://regis.edu/Academics/Academic-Grants/Proposals/Regis-Information/IRB.aspx>.

The CC&IS Policies & Procedures Syllabus Addendum summarizes additional important policies including, Diversity, Equal Access, Disability Services, and Attendance & Participation that apply to every course offered by the College of Computer & Information Sciences at Regis University.

A copy of the CC&IS Policies & Procedures Syllabus Addendum can be found here: <https://in2.regis.edu/sites/ccis/policies/Repository/CCIS%20Syllabus%20Addendum.docx>.