

**STATE UNIVERSITY OF NEW YORK  
COLLEGE OF TECHNOLOGY  
CANTON, NEW YORK**



**MASTER SYLLABUS**

**CITA 385 – COBOL for Business and Accounting**

**Created by: Robert House  
Updated by: Minhua Wang**

**CANINO SCHOOL OF ENGINEERING TECHNOLOGY  
DECISION SYSTEMS  
FALL 2018**

- A. **TITLE:** Visual Programming and Development Tools
- B. **COURSE NUMBER:** CITA 385
- C. **CREDIT HOURS:** (Hours of Lecture, Laboratory, Recitation, Tutorial, Activity)

# Credit Hours: 3  
 # Lecture Hours: 2 per week  
 # Lab Hours: 1 (two-hour lab) per week  
 Other: per week

Course Length: 15 Weeks

- D. **WRITING INTENSIVE COURSE:** No
- E. **GER CATEGORY:** None
- F. **SEMESTER(S) OFFERED:** As required
- G. **COURSE DESCRIPTION:** This course provides students with the knowledge and experience to write and modify programs written in the COBOL programming language. Classroom exercises use real world scenarios so students will gain an understanding of where COBOL fits in the business world.
- H. **PRE-REQUISITES/CO-REQUISITES:**
- a. Pre-requisite(s): CITA 152 Computer Logic
  - b. Co-requisite(s): none
  - c. Pre- or co-requisite(s): none

I. **STUDENT LEARNING OUTCOMES:**

By the end of this course, the student will be able to:

| <b><u>Course Student Learning Outcome</u></b><br><b><u>[SLO]</u></b> | <b><u>PSLO</u></b>  | <b><u>ISLO</u></b> |
|--|---|--------------------|
| a. Design algorithms to solve application problems                   | 3. Demonstrate a solid understanding of the methodologies and foundations of IT | 5                  |
| b. Compile, test, and debug COBOL programs                           | 3. Demonstrate a solid understanding of the methodologies and foundations of IT | 5                  |
| c. Define data elements used in a COBOL program                      | 3. Demonstrate a solid understanding of the methodologies and foundations of IT | 5                  |
| d. Create procedures using COBOL statements                          | 3. Demonstrate a solid understanding of the methodologies and foundations of IT | 5                  |
| e. Read, modify, and write data files using COBOL                    | 3. Demonstrate a solid understanding of the methodologies and foundations of IT | 5                  |

J. **APPLIED LEARNING COMPONENT:** Yes X No \_\_\_\_\_

- Classroom/Lab

K. % **TEXTS:** *Murach's Mainframe COBOL*. Mike Murach, Anne Prince, and Raul Menendez. ISBN: 978-1-890774-24-0.

L. % **REFERENCES:** *COBOL for the 21st Century*. Nancy Stern, Robert Stern, and James Ley. ISBN: 978-1-118-73953-1.

M. % **EQUIPMENT:** Computer lab classroom

N. % **GRADING METHOD:** A-F

O. % **SUGGESTED MEASUREMENT CRITERIA/METHODS:**

- Exams
- Projects
- Assignments

P. **DETAILED COURSE OUTLINE:**

- I. Introduction to COBOL programming
  - A. Compiling, testing, and debugging a COBOL program
  - B. Structured program
    1. Designing
    2. Structured coding
    3. Testing
  - C. Other ways to define, move, and initialize fields
  - D. Other COBOL essentials
    1. Arithmetic statements
    2. Intrinsic functions
    3. Working with dates
    4. Working with characters
    5. Working with tables
    6. Using copy members and subprograms
- II. COBOL for working with files
  - A. Working with disk files
    1. Sequential files
    2. Indexed files
  - B. Using the sort/merge feature
- III. The IBM mainframe environment for COBOL
  - A. Introduction to IBM mainframes
  - B. Compiling and testing programs on an IBM mainframe
  - C. Using CICS to develop interactive programs
  - D. Using DB2 to develop database programs
  - E. How to become an effective maintenance programmer

**Q.    LABORATORY OUTLINE:**

Labs will be several projects which must be designed and coded using COBOL statements. The number of lab days required will depend on the specific project. Projects that may be assigned include, but are not limited to, the following:

1. Calculate grade point average
2. Calculate totals and averages
3. Prepare a student listing
4. Prepare an honor student report
5. Prepare a student roster
6. Prepare a multi-line student roster
7. Prepare a student roster summary
8. Prepare a two-level student registration report
9. Prepare a three-level student registration summary
10. Apply packed-decimal or binary usage to an old program
11. Develop a number guessing game
12. Prepare a student age listing
13. Project graduation age and days until graduation
14. Translate English to Pig Latin
15. Unstring name, number, and password
16. Use a state table to validate states and zip codes
17. Prepare GPA range summary
18. Use copy members
19. Create a subprogram that converts numbers to words
20. Create a subprogram that unstrings name, number, and password
21. Create a subprogram for state and zip code validation
22. Prepare a student registration report
23. Update the student master file
24. Maintain the student master file
25. Create an indexed student master file
26. Prepare student registration report
27. Update the student master file
28. Prepare a student listing in descending GPA sequence