



Broward Community College

Course Outline

STATUS: A

COMMON COURSE NUMBER: COP 1120

COURSE TITLE: COBOL I

CREDIT HOURS: 3

CONTACT HOURS BREAKDOWN:

Lecture/Discussion 48

Lab 16

Other

Contact Hours/Week 3

CATALOG COURSE DESCRIPTION:

Prerequisite: CGS 1000 or Instructor's Approval

Corequisite: None

The COBOL programming language is taught in a structured format. Through lectures and laboratory practices, students develop program-ming ability in the COBOL language. Emphasis is placed on sequential file processing and the creation of different reports. Creation of disk files is also covered.

General Education Requirements - Associate of Arts Degree, meets Area(s):
General Education Requirements - Associate in Science Degree, meets Area(s):

UNIT TITLES:

1. Introduction
2. Problem Solving and Planning Tools
3. Introduction to COBOL
4. Input and Output
5. Arithmetic Operations
6. Structured Logic
7. Decision Making
8. Disk Processing
9. Advanced Data Field Definitions
10. Table Processing

LAST REVIEW Academic Year 2004-2005 NEXT REVIEW Academic Year 2009-2010

Interim Revision Dates: 10/90, 10/92

I. Course Overview:

Upon successful completion of this course, the students should be able to design the logic, code, test, debug, and execute in COBOL at an introductory level.

II. Units:

Unit 1. Introduction

General Outcome:

1.0 The students should be able to relate basic computer components.

Specific Learning Outcomes:

Upon successful completion of this unit, the students should be able to:

- 1.1 Identify and describe components of a computer.
- 1.2 Identify and describe types of computer languages.
- 1.3 Describe the compilation process.
- 1.4 Define computer terminology.

Unit 2. Problem Solving and Planning Tools

General Outcome:

2.0 The students should be able to explain the necessary steps for planning problem solutions.

Specific Learning Outcomes:

Upon successful completion of this unit, the students should be able to:

2.1 Identify and describe the steps taken in problem-solving.

2.2 Identify and describe elements in flowcharting and logic planning.

2.3 Discuss the use of flowcharting to plan the logic of a program.

2.4 Describe other logic planning tools.

2.4.1 Structured design

2.4.1.1 hierarchy chart

2.4.1.2 IPO charts

2.4.1.3 pseudocode

2.4.2 Decision tables

Unit 3. Introduction to COBOL

General Outcome:

3.0 The students should be able to discuss the overall format and organization of a COBOL program.

Specific Learning Outcomes:

Upon successful completion of this unit, the students should be able to:

3.1 Name and describe the four divisions of COBOL.

3.2 Differentiate between Area A and Area B.

Unit 4. Input and Output

General Outcome:

4.0 The students should be able to describe and code the input data and the computer output it will produce.

Specific Learning Outcomes:

Upon successful completion of this unit, the students should be able to:

4.1 Describe the Data Division.

4.2 Identify types of data.

4.2.1 Alphabetic

4.2.2 Numeric

4.2.3 Alphameric

4.3 Define basic data structure.

4.3.1 Field

4.3.2 Record

4.3.3 File

4.4 Define and develop record layouts for input and output.

4.4.1 File and field names

4.4.2 Size of fields

4.5 Describe sequential file processing.

4.6 Develop titles and headings.

Unit 5. Arithmetic Operations

General Outcome:

5.0 The students should be able to select the correct verb and code the correct fashion to solve an arithmetic equation.

Specific Learning Outcomes:

Upon successful completion of this unit, the students should be able to:

- 5.1 Apply the verbs ADD, SUBTRACT, MULTIPLY, DIVIDE, and COMPUTE.
- 5.2 Describe the purpose of the ROUNDED and ON SIZE ERROR phrases.
- 5.3 Utilize counters and accumulators.

Unit 6. Structured Logic

General Outcome:

6.0 The students should be able to apply and order logic in a structured fashion.

Specific Learning Outcomes:

Upon successful completion of this unit, the students should be able to:

- 6.1 Discuss the purpose of the PERFORM command.
- 6.2 Discuss the purpose of a PERFORM thru.
- 6.3 Discuss the purpose of a PERFORM thru . . . until.
- 6.4 Apply the concept of structured programming.

Unit 7. Decision Making

General Outcome:

7.0 The students should be able to explain the use of the IF command for conditional branching.

Specific Learning Outcomes:

Upon successful completion of this unit, the students should be able to:

7.1 Describe the purpose of the IF statement.

7.2 Describe and use the relational test.

7.3 Describe and apply the class test.

7.4 Describe and apply level 88 condition names.

7.5 Describe and utilize the compound and nested IF.

7.6 Describe the responsibility of error checking in a program.

Unit 8. Disk Processing

General Outcome:

8.0 The students should be able to explain how files are saved to a disk and accessed from a disk.

Specific Learning Outcomes:

Upon successful completion of this unit, the students should be able to:

8.1 Contrast disk and tape processing.

8.2 Describe disk processing.

8.3 Record and retrieve data using a sequential file.

Unit 9. Advanced Data Field Definitions

General Outcome:

9.0 The students should be able to expand the concepts of data field manipulation and definition.

Specific Learning Outcomes:

Upon successful completion of this unit, the students should be able to:

9.1 Describe and utilize the redefines to change a data image.

9.2 Describe and utilize the redefines to save storage.

Unit 10. Table Processing

General Outcome:

10.0 The students should be able to describe and code one-level tables.

Specific Learning Outcomes:

Upon successful completion of this unit, the students should be able to:

10.1 Use tables in a variety of business problems.

10.2 Identify and define the following terms:

- 10.2.1 Subscript
- 10.2.2 Single dimension array
- 10.2.3 Table
- 10.2.4 Occurs clause
- 10.2.5 Redefines clause
- 10.2.6 Perform varying
- 10.2.7 Sequential search
- 10.2.8 Binary search

10.3 Create an internal table within the program.

10.4 Code routines for table hook-up and table search.

Special Student Projects:

The students will be assigned six to eight (6-8) problems as laboratory programming exercises. These exercises will be, for the most part, prepared by the students during their assigned lab period, plus other outside-of-class time.

Each exercise may include the following (at the instructor's discretion):

- an algorithm design
- a program pseudocode
- a flowchart
- a structure chart
- program coding
- a screen and print report design
- a program listing
- a program output