



# Broward Community College

## Course Outline

STATUS:   A  

COMMON COURSE NUMBER:   COP 1170  

COURSE TITLE:   Programming in BASIC  

CREDIT HOURS:           3          

**CONTACT HOURS BREAKDOWN:**

Lecture/Discussion           48          

Lab                   16          

Other                           

Contact Hours/Week           3          

**CATALOG COURSE DESCRIPTION:**

Prerequisite: MAT 0024 or higher

Corequisite: CGS 1000

This course provides a study of programming utilizing the BASIC language. Emphasis is placed on the development of computer problem-solving skills and structured programming techniques in business, engineering, mathematics, science, and other related fields. Lectures and discussions are supplemented by assigned laboratory work.

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

**UNIT TITLES:**

1. Algorithms/Flowcharting
2. Input/Output
3. Arithmetic Operations
4. Loop Control Structures
5. Decision-Making Structures
6. Arrays
7. Functions/Subroutines
8. String Manipulation
9. Files
10. Variable Types

## **I. Course Overview:**

Upon successful completion of this course, the students should be able to design the logic, code, test, debug, and execute BASIC language programs.

## **II. Units:**

### **Unit 1. Algorithms/Flowcharting**

#### General Outcome:

- 1.0 The students should be able to develop algorithms and use a logic tool to express the algorithms.

#### Specific Learning Outcomes:

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

- 1.1 Formulate algorithms for a variety of problems.
- 1.2 Use graphical representation and other programming tools to represent the algorithm in a form conducive to programming.

## Unit 2. Input/Output

### General Outcome:

2.0 The students should be able to manipulate BASIC statements governing the format in which a program inputs data and outputs processed information.

### Specific Learning Outcomes:

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

2.1 Write READ, INPUT, and PRINT statements using the appropriate BASIC statement to produce the desired input or output of data.

2.2 Round numeric values, justify, insert commas, dollar signs, asterisks, blanks, and other printing characters at precise locations within a line of output.

2.3 Create printer spacing charts to guide the placement and appearance of the output.

### Unit 3. Arithmetic Operations

#### General Outcome:

3.0 The students should be able to perform calculations using the assignment statement by applying computer arithmetic operators.

#### Specific Learning Outcomes:

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

3.1 Write assignment statements that perform arithmetic calculations.

3.2 Apply the hierarchy of operation symbols and the use of parentheses to control the sequence of mathematical operations.

3.3 Apply pre-defined functions to perform arithmetic operations.

## Unit 4. Loop Control Structures

### General Outcome:

4.0 The students should be able to control the repetition of a statement block using a loop control construct.

### Specific Learning Outcomes:

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

- 4.1 Use a pre-test loop construct to control repetition of a statement block.
- 4.2 Use a post-test loop construct to control repetition of a statement block.
- 4.3 Use a counter-controlled loop construct to control repetition of a statement block.

## Unit 5. Decision-Making Structures

### General Outcome:

5.0 The students should be able to control the choice of statement block to execute using a decision-making control structure.

### Specific Learning Outcomes:

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

5.1 Use relational and logical operators to construct Boolean expressions that evaluate to either TRUE or FALSE.

5.2 Use the IF-THEN-ELSE construct to select an appropriate statement block to execute.

5.3 Use the SELECT CASE construct to select an appropriate statement block to execute.

## Unit 6. Arrays

### General Outcome:

6.0 The students should be able to declare an array size and apply arrays in a BASIC program.

### Specific Learning Outcomes:

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

- 6.1 Dimension different size arrays.
- 6.2 Create subscripted variables names to symbolize the content of an array element.
- 6.3 Apply one-, two-, and three-dimension arrays in a BASIC program.
- 6.4 Execute indexed arrays for input and output.
- 6.5 Search an array using sequential and binary search techniques.
- 6.6 Sort a one-dimension array in ascending or descending order.
- 6.7 Use parallel arrays to perform table lookup operations.

## Unit 7. **Functions/Subroutines**

### General Outcome:

7.0 The students should be able to create and/or apply functions and subroutines.

### Specific Learning Outcomes:

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

- 7.1 Apply pre-defined functions.
- 7.2 Create and apply user-defined functions.
- 7.3 Create and apply subroutines.
- 7.4 Pass arguments to functions or subroutines.
- 7.5 Specify the difference between passing arguments by value versus passing arguments by reference.

## Unit 8. String Manipulation

### General Outcome:

8.0 The students should be able to input, print, and manipulate string data.

### Specific Learning Outcomes:

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

- 8.1 Input string data via the INPUT and READ/DATA statements.
- 8.2 Compare character string data.
- 8.3 Represent the numerical code of characters.
- 8.4 Manipulate individual characters or subsets of strings using string functions.
- 8.5 Print string output.

## Unit 9. Files

### General Outcome:

9.0 The students should be able to create, read, and manipulate sequential and random files.

### Specific Learning Outcomes:

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

9.1 Apply the OPEN statement for sequential and random files.

9.2 Apply the end of file function.

9.3 Output to and input from sequential and/or random files.

9.4 Update a random file.

9.5 Apply the CLOSE statement for sequential and random files.

## Unit 10. Variable Types

### General Outcome:

10.0 The students should be able to classify data according to data types and explain how this classification affects the use of the data in a program.

### Specific Learning Outcomes:

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

10.1 Apply numerical variable names.

10.2 Apply string variable names.

10.3 Apply array variable names for numeric or string data.

### Special Student Projects:

The students will be assigned five to twelve (5-12) 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