



## Course Number and Title: CIS 120 - Introduction to Programming

**Campus Location:**

Georgetown, Dover, Wilmington

**Effective Date:**

2018-51

**Prerequisite:**

Test score or ENG 090 or higher, MAT 020 or higher, SSC 100 or concurrent

**Co-Requisites:**

none

**Course Credits and Hours:**

4.00 credits

3.00 lecture hours/week

2.00 lab hours/week

**Course Description:**

This course provides students with an introduction to the design and implementation of basic computer programming. Topics include logic development, control structures, variables, input/output, and debugging techniques of modern programming.

**Required Text(s):**

Obtain current textbook information by viewing the [campus bookstore - https://www.dtcc.edu/bookstores](https://www.dtcc.edu/bookstores) online or visit a campus bookstore.

Check your course schedule for the course number and section.

**Additional Materials:**

USB flash drive for lab work and broadband Internet access

**Schedule Type:**

Classroom Course

Online Course

**Disclaimer:**

None

**Core Course Performance Objectives (CCPOs):**

1. Translate logic into computer executable instructions. (CCC 2, 6; PGC 3, 4)
2. Evaluate and construct control structures of computer programs. (CCC 2, 6; PGC 3, 4)
3. Construct an effective computer program using proper variables. (CCC 2, 6; PGC 3, 4)
4. Justify the use of modularization in computer programming. (CCC 1, 2, 6; PGC 3, 4)
5. Construct user-friendly input and output for effective user communication. (CCC 1, 2, 4,6; PGC 3, 4)
6. Apply debugging techniques to analyze, identify, and repair program errors. (CCC 2, 6; PGC 3, 4)

See Core Curriculum Competencies and Program Graduate Competencies at the end of the syllabus. CCPOs are linked to every competency they develop.

**Measurable Performance Objectives (MPOs):**

Upon completion of this course, the student will:

1. Translate logic into computer executable instructions.
  1. Define conditional programming logic.
  2. Describe a solution to real-world problem.
  3. Identify the steps to solve a real-world problem.
  4. Develop logic to solve a real-world problem.
  5. Evaluate and validate logic developed to solve a real-world problem.
  6. Implement logic into computer code.
2. Evaluate and construct control structures of computer programs.
  1. Define control structures.
  2. Compare and contrast sequence, selection, and loop control structures.
  3. Select the appropriate control structure(s) based on a given condition.
  4. Demonstrate the appropriate use of control structures.
3. Construct an effective computer program using proper variables.
  1. Define variables.
  2. Describe the relationship between proper variable name use, documentation, and effective computer programming.
  3. Compare and contrast the different types of variables and their sizes.
  4. Select appropriate variables to develop an effective computer program.
  5. Analyze the effect of storing values in different variables to develop an effective computer program.
  6. Define, name, and manipulate array elements.
4. Justify the use of modularization in computer programming.
  1. Define functions, procedures, and subroutines.
  2. Define parameters in functions, procedures, and subroutines.
  3. Demonstrate the appropriate use of parameter passing.
  4. Compare and use variables in different scopes to develop an effective computer program.
  5. Apply effective use of modularization in computer programming.
5. Construct user-friendly input and output for effective user communication.
  1. Formulate proper user prompting for input.
  2. Analyze user input for validation.
  3. Compare and contrast different input and output techniques.
  4. Utilize input and output techniques.
  5. Analyze and implement file input and output techniques.
  6. Evaluate the accuracy of output.
6. Apply debugging techniques to analyze, identify, and repair program errors.
  1. Differentiate among the types of errors.
  2. Analyze logic and program errors using debugging techniques.
  3. Develop and implement solution(s) to solve programming errors.
  4. Evaluate the effectiveness of debugging techniques.

**Evaluation Criteria/Policies:**

Students must demonstrate proficiency on all CCPOs at a minimal 75 percent level to successfully complete the course. The grade will be determined using the Delaware Tech grading system:

92	-	100	=	A
83	-	91	=	B
75	-	82	=	C
0	-	74	=	F

Students should refer to the [Student Handbook - https://www.dtcc.edu/handbook](https://www.dtcc.edu/handbook) for information on the Academic Standing Policy, the Academic Integrity Policy, Student Rights and Responsibilities, and other policies relevant to their academic progress.

**Core Curriculum Competencies (CCCs are the competencies every graduate will develop):**

1. Apply clear and effective communication skills.
2. Use critical thinking to solve problems.
3. Collaborate to achieve a common goal.
4. Demonstrate professional and ethical conduct.
5. Use information literacy for effective vocational and/or academic research.
6. Apply quantitative reasoning and/or scientific inquiry to solve practical problems.

**Program Graduate Competencies (PGCs are the competencies every graduate will develop specific to his or her major):**

1. Utilize the latest integrated application software package.
2. Install, configure and secure operating systems and application software.
3. Design, write, and debug structured business computer programs.
4. Analyze and design complex computer applications to solve business problems.
5. Integrate the principles of the Internet into web development.

**Disabilities Support Statement:**

The College is committed to providing reasonable accommodations for students with disabilities. Students are encouraged to schedule an appointment with the campus Disabilities Support Counselor to request an accommodation needed due to a disability. A listing of campus Disabilities Support Counselors and contact information can be found at the [disabilities services - https://www.dtcc.edu/disabilitysupport](https://www.dtcc.edu/disabilitysupport) web page or visit the campus Advising Center.