Introduction to C# Programming Applications

PEIMS Code: N1302812

Abbreviation: INTCPA

Grade Level(s): 11–12

Award of Credit: 1.0

Approved Innovative Course

- Districts must have local board approval to implement innovative courses.
- In accordance with Texas Administrative Code (TAC) §74.27, school districts must provide instruction in all essential knowledge and skills identified in this innovative course.
- Innovative courses may only satisfy elective credit toward graduation requirements.
- Please refer to TAC §74.13 for guidance on endorsements.

Course Description:

Introduction to C# Programming Applications is an introductory programming course using the C# programming language. C# is primarily used on the Windows .NET framework. This course does not assume any prior programming experience, though students should have a working knowledge of Object-Oriented Programming concept. Students will use C# to create several different programs and applications: mobile apps, desktop apps, cloud-based services, websites, enterprise software, and games. In this course, students will explore data types, control structures, functions, syntax, and semantics of the language, classes, class relationships, and exception handling for C#.

Students will demonstrate the use of software development tools. Students will develop and write documented C# programs, including designing, debugging, and analyzing code. The course provides and prepares students to enter the workforce or college with a solid foundation in C# programming. The course prepares students to create different mobile apps, desktop apps, cloud-based services, websites, enterprise software, and games.

Essential Knowledge and Skills:

(a) General Requirements. This course is recommended for students in Grades 11-12. Recommended Prerequisites: At least one credit in a Level 2 or higher course in programming or software development. Students shall be awarded one credit for successful completion of this course.

(b) Introduction.

(1) Career and technical education instruction provide content aligned with challenging academic standards and relevant technical knowledge and skills for students to further their education and succeed in current or emerging professions.

(2) The Information Technology (IT) Career Cluster focuses on building linkages in IT occupations for entry-level, technical, and professional careers related to design, development, support, and management of hardware, software, multimedia, and systems integration services.

(3) In Introduction to C# Programming, students will acquire knowledge of C# syntax including data types, control structures, functions, syntax, and semantics of language, classes, class relations, and exception handling. Students will analyze the social responsibility of business and industry regarding the significant issues relating to the environment, ethics, health, safety, and diversity in society and in the workplace as related to computer programming. Students will apply technical skills to address business applications of emerging technologies.

(4) Students are encouraged to participate in extended learning experiences such as career and technical student organizations and other leadership or extracurricular organizations.

(5) Statements that contain the word "including" reference content that must be mastered, while those containing the phrase "such as" are intended as possible illustrative examples.

(c) Knowledge and Skills.

(1) The student demonstrates professional standards and employability skills as required by the computer programming industry. The student is expected to:

(A) express ideas in a clear, concise, and effective manner;

(B) explain the importance of cooperating, contributing and collaborating as a team member of a team;

(C) describe effective reading and writing skills;

(D) describe effective verbal and nonverbal communication skills;

(E) solve problems and think critically;

(F) explain the importance of leadership skills and function effectively as a team member;

(G) identify and implement proper safety procedures;

(H) describe environmental issues related to the field of information technology (IT);

(I) explain and discuss the relevance of diversity in society and the workplace;

(J) explain legal and ethical responsibilities in relation to the field of IT; and

(K) describe planning and time-management skills such as project management.

(2) The student identifies various employment opportunities in the information technology field. The student is expected to:

(A) identify job opportunities and accompanying job duties and tasks;

(B) investigate emerging and innovative technologies that are potential career opportunities;

(C) identify careers of personal interest along with the education, job skills, and experience required to achieve personal career goals; and

(D) outline the functions of resumes and portfolios.

(3) The student identifies basic concepts and defines terminology associated with computer systems and program development. The student is expected to:

(A) identify and describe appropriate terminologies, such as C# terms, syntax, data types, objects, concepts, purposes, control structures, exceptions, classes, and arrays;

(B) identify and describe various software applications;

(C) compare the various ways that computers and programming languages are used for personal, workgroup, and enterprise computing;

(D) identify and describe multiple logic structures used in software design;

(E) explain the hardware and software aspects of computer systems that support application software development; and

(F) identify the fundamental principles of programming, including those of algorithm analysis, software design, operating systems, and database.

(4) The student demonstrates the use of software development tools and applies problem-solving skills to implement software design. The student is expected to:

(A) explain the general problem-solving concepts and steps used in software design;

(B) apply C# terms, syntax, data types, objects, concepts, purposes, control structures, exceptions, classes, and arrays to software design;

(C) explain the use of procedural programming structure;

(D) describe the use of arrays in solving problems;

(E) apply sequential logic structure in software design; and

(F) apply data structures and algorithms in software design.

(5) The student develops and writes documented C# programs, including designing, debugging and analyzing code. The student is expected to:

(A) explain the use of C# development tools;

(B) choose appropriate data and control structures based on assigned criteria;

(C) explain the use of loops and case logic structures;

(D) apply file and database concepts;

(E) design and develop correct executable projects;

(F) develop C# desktop graphical user interface (GUI) programs;

(G) create appropriate documentation; and

(H) describe and demonstrate debugging and exceptions handling.

Recommended Resources and Materials:

Microsoft Visual Studio Professional 2022

Doyle, B. (2016). C# Programming from Problem Analysis to Program Design, 5th Edition: Cengage Learning

C# projects archives. Source Code & Projects. (2022, July 13). Retrieved January 17, 2023, from https://code-projects.org/c/languages/project/c-sharp-projects/

C# sharp programming exercises, practice, solution. w3resource. (n.d.). Retrieved January 17, 2023, from https://www.w3resource.com/csharp-exercises/

Global, TELCOMA. “C In Depth: The Complete C Programming Guide for Beginners.” TutorialsPoint, February 22, 2021. https://www.tutorialspoint.com/c-in-depth-the-complete-c-programming-guide-for-beginners/index.asp.

“C# Exercises on Exercism.” Exercism. Accessed March 1, 2023. https://exercism.org/tracks/csharp/exercises.

Recommended Course Activities:

- Development of C# desktop GUI programs.
- Development of C# programs that utilize the .NET Collections Framework.
- Application of sequential logic structure.
- Design and development of correct executable projects.
- Creation of appropriate documentation.
- Debugging and analysis of program code.

Suggested methods for evaluating student outcomes:

Performance on assigned projects and teacher-developed assessments will be used to determine the student’s success. Students will be evaluated on skill competency, written tests, daily grades, and group/individual projects using rubrics.

- **Formative Assessments**
  - Class observation
  - Reflections journals
  - Peer Feedback
  - Exit Ticket
- **Summative Assessments**
  - Projects
  - Portfolios
  - Peer Feedback
  - Unit Test

Teacher qualifications:

- Computer Science: Grades 8-12
- Secondary Computer Information Systems: Grades 6-12
- Secondary Industrial Arts: Grades 6-12
- Secondary Industrial Technology: Grades 6-12
- Technology Education: Grades 6-12
- Trade and Industrial Education: Grades 6-12. This assignment requires appropriate work approval.
- Trade and Industrial Education: Grades 8-12. This assignment requires appropriate work approval.
- Vocational Trades and Industry. This assignment requires appropriate work approval.

Additional information:

N/A