eDynamic Learning Course Title: Programming 2a/2b

State: TX
State Course Title: Computer Programming II
State Course Code: 130.310
State Standards: Information Technology
Date of Standards: 2015

TEKS	Course Title. (a or b), if applicable, e.g. Game Design 1a	Unit Name(s)	Lesson(s) Numbers
(1) The student demonstrates professional standards / employability skills as required by business and industry.			
(A) employ effective reading and writing skills;	Programming 2a: Procedural Programming	Unit 3: Working With Data	Lesson 4
(B) employ effective verbal and nonverbal communication skills;	Programming 2b: Creative Programming	Unit 5: Communicating in Teams	Lessons 1-4
(C) illustrate interview skills for successful job placement;	Programming 2b: Creative Programming	Unit 8: Go Team	Lessons 1-4
(D) solve problems and think critically;	Programming 2b: Creative Programming	Unit 8: Go Team	Lessons 1-4
(E) demonstrate leadership skills and function effectively as a team member;	Programming 2b: Creative Programming	Unit 5: Communicating in Teams	Critical Thinking 1-5, Activity 1, 2
(F) identify and implement proper safety procedures;	Programming 2b: Creative Programming	Unit 6: Taking Control: Security and Ethics	Activity 2
(G) demonstrate an understanding of legal and ethical responsibilities in relation to the field of IT; and	Programming 2b: Creative Programming	Unit 7: Graphical User Interfaces	Lesson 4
(H) demonstrate planning and time-management skills such as project management, including initiating, planning, executing, monitoring and controlling, and closing a project.	Programming 2a: Procedural Programming	Unit 8: Real-World Inspiration: Capstone Project	Activity 2
(2) The student identifies various employment opportunities in the IT field.			
(A) create a personal career plan along with education, job skills, and experience necessary to achieve career goals; and	Programming 2b: Creative Programming	Unit 8: Go Team	Lessons 1-4
(B) develop a resume that includes letters of recommendation appropriate to a chosen career plan.	Programming 2b: Creative Programming	Unit 8: Go Team	Lessons 1-4
(3) The student identifies project software needs and requirements.			
(A) identify input and output requirements;	Programming 2a: Procedural Programming	Unit 3: Working With Data	Lessons 2-4

(B) identify system processing requirements;	Programming 2a: Procedural Programming	Unit 3: Working With Data	Lesson 1
(C) identify hardware, networking, and software system functional requirements;	Programming 2a: Procedural Programming	Unit 8: Real-World Inspiration: Capstone Project	Activity 2
(D) conduct a project needs analysis;	Programming 2a: Procedural Programming	Unit 3: Working With Data	Lesson 4
(E) define a problem to be solved by a created application;	Programming 2b: Creative Programming	Unit 5: Communicating in Teams	Activity 2
(F) analyze requirement specifications using current approaches;	Programming 2a: Procedural Programming	Unit 8: Real-World Inspiration: Capstone Project	Activity 2
(G) identify project constraints; and	Programming 2a: Procedural Programming	Unit 8: Real-World Inspiration: Capstone Project	Activity 2
(H) use advanced modeling and analysis of functional requirements.	Programming 2a: Procedural Programming	Unit 2: Speaking the Same Language	Lesson 4
(4) The student produces an IT-based strategy and project plan to solve a provided class problem.			
(A) identify key functions and subsystem capabilities of modern software products;	Programming 2a: Procedural Programming	Unit 8: Real-World Inspiration: Capstone Project	Activity 2
(B) identify software resources and individual product risks; and	Programming 2a: Procedural Programming	Unit 8: Real-World Inspiration: Capstone Project	Activity 2
(C) identify software development methodologies.	Programming 2a: Procedural Programming	Unit 8: Real-World Inspiration: Capstone Project	Activity 2
(5) The student demonstrates knowledge of the software development environment.			
(A) apply prototyping techniques;	Programming 2a: Procedural Programming	Unit 8: Real-World Inspiration: Capstone Project	Lesson 2
(B) use appropriate configuration management tools;	Programming 2a: Procedural Programming	Unit 3: Working With Data	Lessons 2-4
(C) apply language-specific programming techniques;	Programming 2a: Procedural Programming	Unit 2: Speaking the Same Language	Lessons 1-4
(D) develop programs using appropriate language;	Programming 2a: Procedural Programming	Unit 2: Speaking the Same Language	Lessons 1-4
(E) apply the appropriate development environment for each selected language such as the compiler, debugger, test generator, and analyzer;	Programming 2a: Procedural Programming	Unit 6: What's the Big Objective?	Lessons 1-4
(F) use appropriate modeling and analysis tools; and	Programming 2a: Procedural Programming	Unit 3: Working With Data	Lesson 4
(G) use appropriate requirement tracking tools.	Programming 2b: Creative Programming	Unit 5: Communicating in Teams	Activity 2
(6) The student demonstrates knowledge of the software development process.			

(A) articulate the information system life cycle;	Programming 2a: Procedural Programming	Unit 8: Real-World Inspiration: Capstone Project	Activity 2
(B) identify system analysis issues related to design, testing, implementation, and maintenance;	Programming 2a: Procedural Programming	Unit 2: Speaking the Same Language	Lesson 4
(C) identify the use of program design tools in a software-development process; and	Programming 2a: Procedural Programming	Unit 8: Real-World Inspiration: Capstone Project	Activity 2
(D) identify current information life cycle models.	Programming 2a: Procedural Programming	Unit 8: Real-World Inspiration: Capstone Project	Activity 2
(7) The student designs a software application.			
(A) apply principals of system design such as structured, object-oriented, and event-driven processes;	Programming 2a: Procedural Programming	Unit 6: What's the Big Objective?	Lessons 1-3
(B) develop a logical design;	Programming 2a: Procedural Programming	Unit 2: Speaking the Same Language	Lessons 1-4
(C) document design specifications according to a defined procedure;	Programming 2a: Procedural Programming	Unit 2: Speaking the Same Language	Lesson 4
(D) design system input, output, processing, and interfaces;	Programming 2a: Procedural Programming	Unit 3: Working With Data	Lessons 2-4
(E) identify the characteristics and uses of data processing such as batch, interactive, event driven, and object oriented;	Programming 2a: Procedural Programming	Unit 3: Working With Data	Lesson 1
(F) explain algorithmic and data structure concepts;	Programming 2a: Procedural Programming	Unit 3: Working With Data	Lesson 1
(G) identify constraints; (H) identify modular design concepts;	Programming 2a: Procedural Programming	Unit 8: Real-World Inspiration: Capstone Project	Activity 2
(I) identify the features, functions, and architectures of client-server computing;	Programming 2a: Procedural Programming	Unit 2: Speaking the Same Language	Lesson 4
(J) articulate database-management concepts;	Programming 2a: Procedural Programming	Unit 3: Working With Data	Lesson 1
(K) define the objectives of a client-server application;	Programming 2b: Creative Programming	Unit 5: Communicating in Teams	Activity 2
(L) design static and dynamic online processing systems; and	Programming 2b: Creative Programming	Unit 5: Communicating in Teams	Activity 2
(M) employ interface techniques.	Programming 2b: Creative Programming	Unit 5: Communicating in Teams	Activity 2
(8) The student codes a software application.			
(A) apply programming language concepts;	Programming 2a: Procedural Programming	Unit 2: Speaking the Same Language	Lessons 1-4
(B) identify the hardware-software connection;	Programming 2a: Procedural Programming	Unit 8: Real-World Inspiration: Capstone Project	Activity 2

(C) articulate the concept of data representation;	Programming 2a: Procedural Programming	Unit 3: Working With Data	Lesson 1
(D) apply structured, object-oriented, and event-driven programming techniques;	Programming 2a: Procedural Programming	Unit 6: What's the Big Objective?	Lessons 1-3
(E) articulate how a programming language can support multitasking and exception handling;	Programming 2a: Procedural Programming	Unit 2: Speaking the Same Language	Lessons 1-4
(F) identify how current key programming languages work in different operating system environments;	Programming 2a: Procedural Programming	Unit 2: Speaking the Same Language	Lessons 1-4
(G) translate data structures and program design into code in an appropriate language;	Programming 2a: Procedural Programming	Unit 3: Working With Data	Lesson 1
(H) demonstrate key constructs and commands specific to a language;	Programming 2a: Procedural Programming	Unit 2: Speaking the Same Language	Lessons 1-4
(I) identify current programming languages used in software development;	Programming 2a: Procedural Programming	Unit 2: Speaking the Same Language	Lessons 1-4
(J) explain how to resolve program implementation issues such as debugging, documentation, and auditing;	Programming 2a: Procedural Programming	Unit 2: Speaking the Same Language	Lesson 4
(K) articulate software development issues such as correctness, reliability, and productivity;	Programming 2a: Procedural Programming	Unit 2: Speaking the Same Language	Lesson 4
(L) explain code analysis issues related to design, testing, implementation, and maintenance;	Programming 2a: Procedural Programming	Unit 3: Working With Data	Lesson 4
(M) demonstrate how to design and implement programs in a top-down manner;	Programming 2a: Procedural Programming	Unit 8: Real-World Inspiration: Capstone Project	Activity 2
(N) demonstrate how to translate algorithmic and modular design into computer code;	Programming 2a: Procedural Programming	Unit 8: Real-World Inspiration: Capstone Project	Activity 2
(O) explain how programming control structures are used to verify correctness;	Programming 2a: Procedural Programming	Unit 2: Speaking the Same Language	Lessons 2, 3
(P) compile and debug computer code; and	Programming 2a: Procedural Programming	Unit 8: Real-World Inspiration: Capstone Project	Activity 2
(Q) prepare appropriate commenting within code.	Programming 2a: Procedural Programming	Unit 8: Real-World Inspiration: Capstone Project	Activity 2
(9) The student demonstrates knowledge of software testing.			
(A) develop a test plan;	Programming 2a: Procedural Programming	Unit 8: Real-World Inspiration: Capstone Project	Activity 2
(B) define test procedures;	Programming 2a: Procedural Programming	Unit 8: Real-World Inspiration: Capstone Project	Activity 2
(C) develop test cases; and	Programming 2a: Procedural Programming	Unit 8: Real-World Inspiration: Capstone Project	Activity 2
(D) perform software testing.	Programming 2a: Procedural Programming	Unit 8: Real-World Inspiration: Capstone Project	Activity 2

(10) The student performs quality assurance testing.			
(A) explain the software quality assurance process;	Programming 2a: Procedural Programming	Unit 8: Real-World Inspiration: Capstone Project	Activity 2
(B) apply standard requirements for software quality assurance;	Programming 2a: Procedural Programming	Unit 8: Real-World Inspiration: Capstone Project	Activity 2
(C) perform software quality assurance tasks to determine a quality software product; and	Programming 2a: Procedural Programming	Unit 8: Real-World Inspiration: Capstone Project	Activity 2
(D) conduct code inspection.	Programming 2a: Procedural Programming	Unit 8: Real-World Inspiration: Capstone Project	Activity 2
(11) The student applies procedures for maintaining the security of computerized information.			
(A) identify risks to information systems facilities, data, communication systems, and applications;	Programming 2b: Creative Programming	Unit 6: Taking Control: Security and Ethics	Activity 2
(B) comply with federal and state legislation pertaining to computer crime, fraud, and abuse;	Programming 2b: Creative Programming	Unit 7: Graphical User Interfaces	Lesson 4
(C) identify and select controls for information systems facilities, data communications, and applications appropriate to specific risks; and	Programming 2a: Procedural Programming	Unit 2: Speaking the Same Language	Lessons 2, 3
(D) apply procedures used to recover from situations such as system failure and computer virus.	Programming 2b: Creative Programming	Unit 6: Taking Control: Security and Ethics	Lessons 2-4