

# INFORMATION TECHNOLOGY CAREER CLUSTER DESIGN

## Programming and Software Development Pathway – CIP Code 11.0201

### Approved Pathway:

- 1) Includes minimum of three secondary-level credits.
- 2) Includes a work-based element.
- 3) Consists of a sequence:  
Introductory-level,  
Technical-level, and  
Application-level  
courses.
- 4) Supporting documentation includes Articulation Agreement(s), Certification, Program Improvement Plan, and a Program of Study.
- 5) Technical-level and Application-level courses receive .5 state-weighted funding in an approved CTE pathway.

### **INTRODUCTORY LEVEL**

Computing Systems	10002/60002	1 credit
Computer Applications	10004/60004	1 credit
Intro. To Computer Coding	31001	.5 credit
Intro to Physical Computing	31002	.5 credit

### **TECHNICAL LEVEL**

Technical introduction to Computer Science	41010	1 credit
Computer Programming	10152	1 credit
Database Applications	10053	1 credit
Web Page Design	10201	1 credit
Data Systems/Processing	10054	1 credit
Computer Prog Othr Lang	10156	1 credit
AP Computer Science Principles	31094	1 credit

### **APPLICATION LEVEL**

Cyber Security	41036	1 credit
AP Computer Science A	10157	1 credit
IB Computing	10159	1 credit
Particular Topics in Computer Prog.	10160	1 credit
Programming and Software Development	31098	1 credit
Project Management		

# **KANSAS STATE CAREER CLUSTER COMPETENCY PROFILE**

## **INFORMATION TECHNOLOGY CLUSTER**

PROGRAMMING AND SOFTWARE DEVELOPMENT PATHWAY (C.I.P. 11.0201)

**3 - Proficient Achievement**

**2 - Limited Achievement**

**1 - Inadequate Achievement**

**0 - No Exposure**

Graduation Date \_\_\_\_\_

**I certify that the student has received training in the areas indicated.**

Instructor Signature \_\_\_\_\_

Instructor Signature \_\_\_\_\_

Instructor Signature \_\_\_\_\_

Instructor Signature \_\_\_\_\_

### **COMMON CAREER TECHNICAL CORE – CAREER READY STANDARDS**

1. Act as a responsible and contributing citizen and employee
2. Apply appropriate academic and technical skills
3. Attend to personal health and financial well-being
4. Communicate clearly, effectively and with reason
5. Consider the environmental, social and economic impacts of decisions
6. Demonstrate creativity and innovation
7. Employ valid and reliable research strategies
8. Utilize critical thinking to make sense of problems and persevere in solving them.
9. Model integrity, ethical leadership and effective management
10. Plan education and career path aligned to personal goals
11. Use technology to enhance productivity
12. Work productively in teams while using cultural/global competence

### **COMMON CAREER TECHNICAL CORE – INFORMATION TECHNOLOGY CLUSTER STANDARDS**

1. Demonstrate effective professional communication skills and practices that enable positive customer relationships.
2. Use product or service design processes and guidelines to produce a quality information technology product or service.
3. Demonstrate the use of cross- functional teams in achieving IT project goals.
4. Demonstrate positive cyber citizenry by applying industry accepted ethical practices and behaviors.
5. Explain the implications of IT on business development.
6. Describe trends in emerging and evolving computer technologies and their influence on IT practices.
7. Perform standard computer backup and restore procedures to protect IT information.
8. Recognize and analyze potential IT security threats to develop and maintain security requirements.

9. Describe quality assurance practices and methods employed in producing and providing quality IT products and services.
10. Describe the use of computer forensics to prevent and solve information technology crimes and security breaches.
11. Demonstrate knowledge of the hardware components associated with information systems.
12. Compare key functions and applications of software and determine maintenance strategies for computer systems.

## **INTRODUCTORY LEVEL COURSES**

### **10002 Computing Systems**

*Computing Systems courses offer a broad exploration of the use of computers in a variety of fields. These courses have a considerable range of content, but typically include the introduction of robotics and control systems, computer-assisted design, computer-aided manufacturing systems, and other computer technologies as they relate to industry applications.*

#### **3 2 1 0 1. Overview of Systems**

- a. Identify computer classifications and hardware.
  - i. Identify types of computer storage devices.
  - ii. Identify major hardware components and their functions.
  - iii. Identify the different types of computing devices.
- b. Identify new IT technologies and assess their potential importance and impact on the future.
- c. Identify new & emerging drivers and inhibitors of information technology change.
- d. Operate computer-driven equipment and machines.
- e. Apply knowledge of operating systems principles to ensure optimal functioning of system.
- f. Understand data communications trends and issues.
- g. Demonstrate knowledge of data transmission codes and protocols.
- h. Understand elements and types of information processing. (i.e., input, process, output). (e.g., batch, interactive, event-driven, object-oriented).

#### **3 2 1 0 2. Computer Operations**

- a. Identify and understand the fundamentals of operating systems and their components.
- b. Configure systems to provide optimal system interfaces
- c. Clearly document step-by-step installation procedures for future use and configuration.
- d. Apply concepts of privileged instructions and protected mode programming.
- e. Configure peripheral device drivers (e.g., disk, display, printer, modem, keyboard, and mouse).
- f. Allocate disk space, non-sharable resources, and I/O devices.
- g. Interface peripheral devices/controllers in the computer system (e.g., software and hardware interrupts, exceptions, Direct Memory Addressing [DMA], bus structures).
- h. Identify standards and issues related to I/O programming and design of I/O interfaces.
- i. Define hardware-software interface issues for a computer system.
- j. Review automated scheduling software and Identify scheduling priority in programming.
- k. Document procedures and actions through development of audit trails.

### 3 2 1 0 **3. Hardware**

- a. Ensure that hardware and software system components are compatible and licensed prior to performing installation.
- b. Evaluate systems engineering considerations.
- c. Demonstrate knowledge of how bandwidths affect data transmission and on-screen image.
- d. Evaluate information systems problem-solving techniques and approaches.
- e. Determine the accuracy and completeness of the information gathered.
- f. Explain data communications procedures, equipment and media.
- g. Explain measurement techniques for increased productivity due to information systems implementation.
- h. Explain the differences between local and wide area networks.
- i. Explain the benefits of hosting a web site on a local server vs. at an ISP (Internet Service Provider).
- j. Troubleshoot computer-driven equipment and machines and access support as needed (e.g. Test system using diagnostic tools/software, repair/replace malfunctioning hardware and reinstall software as needed, recover data and/or files and restore system to normal operating standards.)

### 3 2 1 0 **4. Software**

- a. Determine software design process, from specification to implementation and appraise software process and product life cycle models.
- b. Explain new and emerging classes of software.
- c. Explain the key functions and applications of software.
- d. Demonstrate knowledge of the function and operation of compilers and interpreters.
- e. Demonstrate knowledge of widely used software applications (e.g., word processing, database management, spreadsheet development, publishing software).
- f. Demonstrate an understanding of various programming paradigms (OO, functional, logic) in software development
- g. Demonstrate knowledge of how data is organized in software development: source version data, project progress data, etc. to increase individual efficiency and respect team member data.
- h. Explain the features and functions of how web browsing software affects the look of a web page, consider the characteristics and uses of plug- ins and examine role of browsers in reading files on the World Wide Web (text-only, hypertext).
- i. Explain the role of number systems in information systems and internal data representation.
- j. Identify the role the binary system in information systems.

### 3 2 1 0 **5. Serving the needs of the end user**

- a. Communicate to understand the problem the user wants to solve independent of the technology (empathy). Consider develop context awareness -- consider the context of the user and the problem before proposing a solution.
- b. Perform software customization as requested to meet the needs of the end user.
- c. Perform installation accurately and completely, using available resources as needed.
- d. Resolve problems with installation if they occur.
- e. Test and maintain products /services.
- f. Initiate predictive maintenance procedures.
- g. Consider customer satisfaction in determining product characteristics (e.g., usefulness, price, operation, life, reliability, safety, and cost of operation).

- h. Use available reference tools (e.g., procedural manuals, documentation, standards, and work flowcharts) as appropriate to access needed information.
- i. Use installation/operation manuals to access needed information using appropriate reference materials.
- j. Use reliability factors (e.g., cost, human, productivity) to plan for and create products/ services; with consideration of maintainability, good design, design simplification, and design redundancy.
- k. Demonstrate knowledge of critical thinking skills, decision-making skills and develop a plan using data-oriented techniques.

## **10004-Computer Applications**

*In Computer Applications courses, students acquire knowledge of and experience in the proper and efficient use of previously written software packages. These courses explore a wide range of applications, including (but not limited to) word-processing, spreadsheet, graphics, and database programs, and they may also cover the use of electronic mail and desktop publishing.*

### **3 2 1 0 1. Personal Information Management**

- a. Identify PIM applications (e.g., Essential PIM, MS Outlook, Lotus Notes...) and maintain safe and secure user profiles.
- b. Manage daily/weekly/monthly schedule using applications such as. (e.g., Notes, MS Outlook, calendars/schedules.)
- c. Create reminder for oneself and send notes/ informal memos using PIM applications.
- d. Access email system using login and password functions. Access email messages received
- e. Create and send e-mail messages in accordance with established business standards (e.g., grammar, word usage, spelling, sentence structure, clarity) demonstrating knowledge of email etiquette.
- f. Attach files to send with messages and access and save received attachments.
- g. Demonstrate knowledge of contamination protection strategies for email.
- h. Maintain shared database of contact information.
- i. Participate in virtual group discussions and meetings.

### **3 2 1 0 2. Research and Internet**

- a. Test Internet connection.
- b. Navigate web sites using software functions. (e.g., Forward, Back, Go To, Bookmarks). Utilize online tools
- c. Explore the multimedia capabilities of the World Wide Web.
- d. Bookmark web addresses (URLs).
- e. Locate information using appropriate search procedures and approaches through a variety of search engines and Boolean logic.
- f. Access, evaluate accuracy, and compile Internet resource information for a variety of purposes. (e.g., library catalogs, business, technical, commercial, government, educational)
- g. Unpack files using compression software. Organize and archive files.

### **3 2 1 0 3. Word Processing and Presentations**

- a. Create/Open Edit and Save documents (e.g., letters, memos, reports) and presentations using existing forms and templates.
- b. Employ word processing utility tools (e.g., spell checker, grammar checker). Locate/replace data using search and replace functions.

- c. Format text using basic formatting functions.
- d. Enhance publications using different fonts, styles, attributes, justification, etc.
- e. Enhance publications using paint/draw functions.
- f. Format new desktop publishing files and recognize the advantages and disadvantages of export options.
- g. Place graphics (e.g., graph, clip art, table) in a document or slide in accordance with basic principles of graphics design and visual communication.
- h. Prepare publications using desktop and cloud publishing applications.

#### 3 2 1 0 **4. Spreadsheets**

- a. Create/Open Edit and Save spreadsheets.
- b. Create charts and graphs from spreadsheets.
- c. Group worksheets.
- d. Input/process data using spreadsheet functions.
- e. Perform calculations using simple formulas.
- f. Locate/replace data using search and replace functions.
- g. Process data using database functions (e.g., structure, format, attributes, relationships, keys).
- h. Perform single- and multiple-table queries (e.g., create, run, save).
- i. Verify accuracy of output.
- j. Maintain shared database of contact information.

#### 3 2 1 0 **6. Ethics and Security**

- a. Demonstrate knowledge of potential internal and external threats to security. Maximize threat reduction.
- b. Assess exposure to security issues.
- c. Demonstrate knowledge of virus protection strategy and ability to load virus detection/protection software.
- d. Identify sources of virus infections and how to remove viruses.
- e. Report viruses in compliance with company standards.
- f. Ensure compliance with security rules, regulations, and codes.
- g. Explore ways to implement countermeasures.
- h. Implement security procedures in accordance with business ethics.
- i. Document security procedures.
- j. Understand how to follow a disaster plan.
- k. Understand how to utilize backup and recovery procedures.
- l. Maintain confidentiality.
- m. Understand how to provide for user authentication (e.g., assign passwords, access level).

#### 3 2 1 0 **7. History / Quality Assurance**

Demonstrate knowledge of the diverse continuous improvement cycles within industry and their characteristics.

(e.g., Baldrige Performance Excellence, Demming, ISO 9000, Six Sigma)

#### 3 2 1 0 **8. Personal Attributes for success. Career Technical Core Skills.**

- a. Act as a responsible and contributing citizen and employee
- b. Demonstrate effective professional communication skills and practices that enable positive customer relationships.
- c. Apply appropriate academic and technical skills
- d. Attend to personal health and financial well-being
- e. Communicate clearly, effectively and with reason

- f. Consider the environmental, social and economic impacts of decisions
  - g. Demonstrate the use of cross-functional teams in achieving IT project goals.
- Demonstrate positive cyber citizenry by applying industry accepted ethical practices and behaviors.

### **31001 Introduction to Computer Coding**

*Students develop an introductory understanding of fundamental computer science concepts and apply computational thinking skills to solve problems through the use of entry level coding tools for computers and/or mobile devices.*

- 3 2 1 0 1. Solicit and integrate peer feedback as appropriate to develop or refine a program.
- 3 2 1 0 2. Compare different algorithms that may be used to solve the same problem, but one might be faster than the other. (e.g., different algorithms solve the same problem, but one might be faster than the other). [Clarification: Students are not expected to quantify these differences.]
- 3 2 1 0 3. Interpret the flow of execution of algorithms and predict their outcomes. [Clarification: Algorithms can be expressed using natural language, flow and control diagrams, comments within code, and pseudocode.]
- 3 2 1 0 4. Design, develop, and present computational artifacts such as mobile applications that address social problems both independently and collaboratively.
- 3 2 1 0 5. Develop programs, both independently and collaboratively, that include sequences with nested loops and multiple branches. [Clarification: At this level, students may use block-based and/or text-based programming languages.]
- 3 2 1 0 6. Create variables that represent different types of data and manipulate their values.
- 3 2 1 0 7. Define and use procedures that hide the complexity of a task and can be reused to solve similar tasks. [Clarification: Students use and modify, but do not necessarily create, procedures with parameters.]
- 3 2 1 0 8. Decompose a problem into parts and create solutions for each part.
- 3 2 1 0 9. Use an iterative design process (e.g., define the problem, generate ideas, build, test, and improve solutions) to solve problems, both independently and collaboratively.
- 3 2 1 0 10. Use a systematic process to identify the source of a problem within individual and connected devices (e.g., follow a troubleshooting flow diagram, make changes to software to see if hardware will work, restart device, check connections, swap in working components).
- 3 2 1 0 11. Explain the processes used to collect, transform, and analyze data to solve a problem using computational tools (e.g., use an app or spreadsheet form to collect data, decide which data to use or ignore, and choose a visualization method.).
- 3 2 1 0 12. Explain how computer science fosters innovation and enhances nearly all careers and disciplines.
- 3 2 1 0 13. Describe ethical issues that relate to computing devices and networks (e.g., equity of access, security and plagiarism), hacking, intellectual property, copyright, Creative Commons licensing.
- 3 2 1 0 14. Summarize security risks associated with weak passwords, lack of encryption, insecure transactions, and persistence of data.

### **31002 Introduction to Physical Computing**

*Students develop an understanding of programming for the physical world, including but not limited to the Internet of Things (IoT), microcontrollers, robotics, and other physical computing devices that are in use in our world.*

- 3 2 1 0 1. Compare different algorithms that may be used to solve the same problem, but one might be

- faster than the other. (e.g., different algorithms solve the same problem, but one might be faster than the other). [Clarification: Students are not expected to quantify these differences.]
- 3 2 1 0 2. Interpret the flow of execution of algorithms and predict their outcomes. [Clarification: Algorithms can be expressed using natural language, flow and control diagrams, comments within code, and pseudocode.]
  - 3 2 1 0 3. Decompose a problem into parts and create solutions for each part.
  - 3 2 1 0 4. Use an iterative design process (e.g., define the problem, generate ideas, build, test, and improve solutions) to solve problems, both independently and collaboratively.
  - 3 2 1 0 5. Analyze the relationship between a device's computational components and its capabilities. [Clarification: computing systems include not only computers, but also cars, microwaves, smartphones, traffic lights, and flash drives.]
  - 3 2 1 0 6. Use a systematic process to identify the source of a problem within individual and connected devices (e.g., follow a troubleshooting flow diagram, make changes to software to see if hardware will work, restart device, check connections, swap in working components).
  - 3 2 1 0 7. Explain the processes used to collect, transform, and analyze data to solve a problem using computational tools (e.g., use an app or spreadsheet form to collect data, decide which data to use or ignore, and choose a visualization method.).
  - 3 2 1 0 8. Represent data using different encoding schemes (e.g., binary, Unicode, Morse code, shorthand, student-created codes).
  - 3 2 1 0 9. Explain how computer science fosters innovation and enhances nearly all careers and disciplines.
  - 3 2 1 0 10. Describe ethical issues that relate to computing devices and networks (e.g., equity of access, security and plagiarism), hacking, intellectual property, copyright, Creative Commons licensing.

## Technical Courses

### 41010 Technical Introduction to Computer Science

*Students develop an understanding of computer science concepts and use computational thinking skills to solve problems through a variety of programming languages and platforms. Students also develop skills in collaboration and teamwork through the use of work models such as pair programming. This course provides a rich technical foundation for continued study*

- 3210 1. Design and develop a software artifact working in a team.
- 3210 2. Compare and contrast various software licensing schemes (e.g., open source, freeware, commercial).
- 3210 3. Design, develop, and implement a computing artifact that responds to an event (e.g., robot that responds to a sensor, mobile app that responds to a text message, sprite that responds to a broadcast).
- 3210 4. Use user - centered research and design techniques (e.g., surveys, interviews) to create software solutions.
- 3210 5. Integrate grade-level appropriate mathematical techniques, concepts, and processes in the creation of computing artifacts.
- 3210 6. Deconstruct a complex problem into simpler parts using predefined constructs (e.g., functions and parameters and/or classes).
- 3210 7. Demonstrate the value of abstraction for managing problem complexity (e.g., using a list instead of discrete variables).
- 3210 8. Design algorithms using sequence, selection, and iteration.
- 3210 9. Use a systematic approach and debugging tools to independently debug a program (e.g., setting breakpoints, inspecting variables with a debugger).
- 3210 10. Convert between binary, decimal, and hexadecimal representations of data (e.g., convert hexadecimal color codes to decimal percentages, ASCII/Unicode representation).



- 3210 11. Debate the social and economic implications associated with ethical and unethical computing practices (e.g., intellectual property rights, hacktivism, software piracy, diesel emissions testing scandal, new computers shipped with malware).
- 3210 12. Compare and debate the positive and negative impacts of computing on behavior and culture (e.g., evolution from hitchhiking to ridesharing apps, online accommodation rental services).
- 3210 13. Compare and contrast multiple viewpoints on cybersecurity (e.g., from the perspective of security experts, privacy advocates, the government).
- 3210 14. Explain the principles of information security (confidentiality, integrity, availability) and authentication techniques.
- 3210 15. Use simple encryption and decryption algorithms to transmit/receive an encrypted message.

## **10152 Computer Programming (eg. Python, Visual Basic, C++, Java, Other Lang.)**

*Computer Programming courses provide students with the knowledge and skills necessary to construct computer programs in one or more languages. Computer coding and program structure are often introduced with the BASIC language, but other computer languages, such as Visual Basic (VB), Java, Pascal, C++, and COBOL, may be used instead. Initially, students learn to structure, create, document, and debug computer programs, and as they progress, more emphasis is placed on design, style, clarity, and efficiency. Students may apply the skills they learn to relevant applications such as modeling, data management, graphics, and text-processing.*

- 3210 1. Summarize the process of IT product and service design.
- 3210 2. Plan for products/services using reliability factors.
- 3210 3. Create products/services using reliability factors.
- 3210 4. Test new products/services for reliability.
- 3210 5. Maintain the reliability of new products/services.
- 3210 6. Identify input and output requirements,
- 3210 7. Identify system processing requirements.
- 3210 8. Define scope of work to meet customer needs.
- 3210 9. Demonstrate knowledge of the key functions and subsystems of the software product.
- 3210 10. Demonstrate knowledge of cross- functional team structures and team members' roles.
- 3210 11. Assess the importance of new technology to future developments.
- 3210 12. Identify data communication trends and major current issues.
- 3210 13. Identify new technologies relevant to information technology.
- 3210 14. Identify system processing requirements.
- 3210 15. Determine compatibility of hardware and software.
- 3210 16. Identify new and emerging classes of software.
- 3210 17. Identify the elements of the information processing cycle (i.e., input, process, output, storage)
- 3210 18. Demonstrate knowledge of software development environment.
- 3210 19. Develop programs using appropriate language.
- 3210 20. Demonstrate knowledge of the information system life cycle.
- 3210 21. Demonstrate knowledge of the concepts of data and procedural representations.
- 3210 22. Demonstrate knowledge of key constructs and commands specific to a language.
- 3210 23. Demonstrate knowledge of how programming control structures are used to verify correctness.

## **10053 Database Applications**

*Database Application courses provide students with an understanding of database development, modeling, design, and normalization. These courses typically cover such*

*topics as SELECT statements, data definition, manipulation, control languages, records, and tables. In these courses, students may use Oracle WebDB, SQL, PL/SQL, SPSS, and SAS and may prepare for certification.*

- 3 2 1 0 1. Develop programs using an appropriate query language.
- 3 2 1 02. Create, populate, and maintain a database.
- 3 2 1 0 3. Use integrated development environment tools to create and modify databases.
- 3 2 1 04 Demonstrate knowledge of relational database management systems.
- 3 2 1 05 Create and explain Entity Relationship Modeling charts and graphs.
- 3 2 1 0 6. Demonstrate knowledge of the basic principles for programming a query language.
- 3 2 1 07. Demonstrate ability to explain query syntax and data retrieval processes.
- 3 2 1 08. Demonstrate knowledge of constructs and operations specific to a query language.
- 3 2 1 0 9. Demonstrate knowledge and usage of query scripting languages (eg, PL/SQL, T-SQL, PL/pgSQL).
- 3 2 1 0 10. Create a database from an entity relationship model using both program code and an integrated development environment.
- 3 2 1 0 11. Perform standard maintenance on the database.
- 3 2 1 012. Create and maintain indexes on databases.
- 3 2 1 0 13. Release software and documentation updates according to procedures.
- 3 2 1 014. Develop scripts and forms that permit access via websites to the database.
- 3 2 1 015. Identify and analyze potential security problems for web access to the database.

### **10201 Web Page Design**

- 3 2 1 01. Develop flowchart, navigational blueprints and schema.
- 3 2 1 02. Create sample design showing placement of buttons/navigational graphics and suggested color scheme.
- 3 2 1 0 3. Develop storyboards.
- 3 2 1 0 4. Demonstrate knowledge of available graphics, video, motion graphics, web software programs.
- 3 2 1 0 5. Identify how different user agents (browsers, devices) affect the digital communication product.
- 3 2 1 0 6. Create and produce content.
- 3 2 1 0 7. Create and refine design concepts.
- 3 2 1 0 8. Identify, utilize and create reusable components.
- 3 2 1 0 9. Apply color theory to select appropriate colors.
- 3 2 1 0 10. Apply knowledge of typography.
- 3 2 1 0 11. Apply principles and elements of design.
- 3 2 1 0 12. Evaluate visual appeal.
- 3 2 1 0 13. Demonstrate knowledge of basic web application security.
- 3 2 1 0 14. Demonstrate knowledge of HTML, XHTML, and CSS.
- 3 2 1 0 15. Explain importance of web standards.
- 3 2 1 0 16. Demonstrate knowledge of Web 2.0.
- 3 2 1 0 17. Explain the importance of ethical behaviors and legal issues.
- 3 2 1 018. Demonstrate knowledge of how to use a scripting language to program a site.
- 3 2 1 0 19. Describe the function of a non-disclosure agreement (NDA).
- 3 2 1 0 20. Differentiate between copyright trademarks.
- 3 2 1 0 21. Explain the concept of intellect property.
- 3 2 1 0 22. Define scope of work to achieve individual and group goals.
- 3 2 1 0 23. Use available reference tools as appropriate.

- 3 2 1 0 24. Explain the features and functions Web browsing software.
- 3 2 1 025. Explain the features and functions of Web design software.
- 3 2 1 026. Compare and contrast clients and servers.
- 3 2 1 027. Describe how bandwidth affects data transmission and on-screen image.

## **10054 Data Systems / Processing**

*Data Systems/Processing courses introduce students to the uses and operation of computer hardware and software and to the programming languages used in business applications. Students typically use BASIC, COBOL, and/or RPL languages as they write flowcharts or computer programs and may also learn data-processing skills.*

- 3 2 1 0 1. Demonstrate knowledge of hard drive technologies (IDE, EIDE, SATA, SCSI, etc).
- 3 2 1 0 2. Demonstrate knowledge of I/O ports (serial, parallel, USB2, USB3, SATA, , etc).
- 3 2 1 0 3. Demonstrate knowledge of INPUT devices (keyboard, mouse, touchpad, cameras, scanners, midis, barcode scanners, etc).
- 3 2 1 0 4. Demonstrate knowledge of OUTPUT devices (printers, CRTs, LCD monitors, network devices).
- 3 2 1 0 5. Demonstrate knowledge of how a programming language can support multitasking and exception- handling.
- 3 2 1 0 6. Demonstrate knowledge of the basic principles for analyzing a programming language.
- 3 2 1 0 7. Demonstrate knowledge of the basics of structured, object-oriented language.
- 3 2 1 0 8. Demonstrate knowledge of the concepts of data and procedural representations.
- 3 2 1 0 9. Demonstrate knowledge of the hardware-software connections.
- 3 2 1 0 10. Demonstrate knowledge of current key programming languages and the environment they are used in.
- 3 2 1 0 11. Demonstrate knowledge of key constructs and commands specific to a language.

## **10156 Computer Programming Other Languages (eg. Python, Visual Basic, C++, Java, Other Lang.)**

- 3210 1. Summarize the process of IT product and service design.
- 32 1 0 2. Plan for products/services using reliability factors.
- 3 2 1 0 3. Create products/services using reliability factors.
- 3 2 1 0 4. Test new products/services for reliability.
- 3 2 1 0 5. Maintain the reliability of new products/services.
- 3 2 1 0 6. Identify input and output requirements,
- 3 2 1 0 7. Identify system processing requirements.
- 3 2 1 0 8. Define scope of work to meet customer needs.
- 3 2 1 0 9. Demonstrate knowledge of the key functions and subsystems of the software product.
- 3 2 1 0 10. Demonstrate knowledge of cross- functional team structures and team members' roles.
- 3 2 1 0 11. Assess the importance of new technology to future developments.
- 3 2 1 0 12. Identify data communication trends and major current issues.
- 3 2 1 0 13. Identify new technologies relevant to information technology.
- 3 2 1 0 14. Identify system processing requirements.
- 3 2 1 0 15. Determine compatibility of hardware and software.
- 3 2 1 0 16. Identify new and emerging classes of software.
- 3 2 1 0 17. Identify the elements of the information processing cycle (i.e., input, process, output, storage)
- 3 2 1 0 18. Demonstrate knowledge of software development environment.

- 3 2 1 0 19 Develop programs using appropriate language.
- 3 2 1 0 20. Demonstrate knowledge of the information system life cycle.
- 3 2 1 0 21. Demonstrate knowledge of the concepts of data and procedural representations.
- 3 2 1 0 22. Demonstrate knowledge of key constructs and commands specific to a language.
- 3 2 1 0 23. Demonstrate knowledge of how programming control structures are used to verify correctness.

## **31094 AP Computer Science Principles**

*AP Computer Science Principles offers a multidisciplinary approach to teaching the underlying principles of computation. The course will introduce students to the creative aspects of programming, abstractions, algorithms, large data sets, the Internet, cybersecurity concerns, and computing impacts. AP Computer Science Principles will give students the opportunity to use technology to address real-world problems and build relevant solutions. Together, these aspects of the course make up a rigorous and rich curriculum that aims to broaden participation in computer science.*

### **Connecting Computing**

- 3210 1. Identify impacts of computing.
- 3210 2. Describe connections between people and computing.
- 3210 3. Explain connections between computing concepts.

### **Developing computational artifacts**

- 3210 4. Create an artifact with a practical, personal, or societal intent.
- 3210 5. Select appropriate techniques to develop a computational artifact.
- 3210 6. Use appropriate algorithmic and information-management principles.

### **Abstracting**

- 3210 7. Explain how data, information, or knowledge are represented for computational use.
- 3210 8. Explain how abstractions are used in computation or modeling.
- 3210 9. Identify abstractions.
- 3210 10. Describe modeling in a computational context.

### **Analyzing problems and artifacts**

- 3210 11. Evaluation of a proposed solution to a problem.
- 3210 12. Location and correction of errors.
- 3210 13. Explanation of how an artifact functions.
- 3210 14. Justification of appropriateness and correctness.

## **APPLICATION LEVEL COURSES**

### **41036 Cybersecurity**

*Students develop an understanding of cybersecurity and information assurance. This includes development of secure systems in terms of hardware, software, and the human-computer interface, including aspects of cryptography and secure computer and network interactions. Successful participants will develop a way of thinking that is security-oriented, better understanding how to think about adversaries and how to build systems that defend against them.*

- 3210 1. Compare and contrast multiple viewpoints on cybersecurity (e.g., from the perspective of security experts, privacy advocates, the government).
- 3210 2. Explain the principles of information security (confidentiality, integrity, availability) and authentication techniques.
- 3210 3. Use simple encryption and decryption algorithms to transmit/receive an encrypted message.
- 3210 4. Identify digital and physical strategies to secure networks and discuss the tradeoffs between ease of access and need for security.
- 3210 5. Explore security policies by implementing and comparing encryption and authentication strategies (e.g., secure coding, safeguarding keys).
- 3210 6. Develop criteria to evaluate the beneficial and harmful effects of computing innovations on people and society.
- 3210 7. Debate laws and regulations that impact the development and use of software.

### **10157 AP Computer Science A**

*Following the College Board's suggested curriculum designed to mirror college-level computer science courses, AP Computer Science A courses provide students with the logical, mathematical, and problem-solving skills needed to design structured, well-documented computer programs that provide solutions to real-world problems. These courses cover such topics as programming methodology, features, and procedures; algorithms; data structures; computer systems; and programmer responsibilities.*

- 3210 1. Demonstrate knowledge of different programming language paradigms. (Eg object oriented programming, functional programming.)
- 3210 2. Read and understand a problem description, purpose, and goals.
- 3210 3. Read and understand class instantiations and relationships among the classes (ie "is-a," and "has-a" relationships).
- 3210 4. Design and implement a class.
- 3210 5. Demonstrate a knowledge and show application of encapsulation.
- 3210 6. Demonstrate a knowledge and show implementation of inheritance.
- 3210 7. Demonstrate a knowledge and show implementation of abstract classes.
- 3210 8. Demonstrate a knowledge of polymorphism.
- 3210 9. Show implementation of method overloading and overriding.
- 3210 10. Understand and implement a given class hierarchy.
- 3210 11. Identify reusable components from existing code using classes and class libraries.
- 3210 12. Demonstrate a proficiency for using libraries as defined by the AP Java Subset.
- 3210 13. Understand and explain the difference between static and dynamic data types.
- 3210 14. Demonstrate a knowledge and show application of recursion.
- 3210 15. Identify edge cases and generate appropriate test data.
- 3210 16. Employ and understand various debugging techniques.
- 3210 17. Demonstrate knowledge of the three types of programming errors: compile or syntax, run-time, and logic.
- 3210 18. Understand error handling and how to throw and catch exceptions.
- 3210 19. Representations of numbers in decimal, binary, and hexadecimal.
- 3210 20. Understanding and implementation of primitive data types int, boolean, and double.
- 3210 21. Demonstrate and show implementation of one dimensional static arrays.
- 3210 22. Demonstrate and show implementation of two dimensional static arrays.
- 3210 23. Knowledge of row-major ordering of two dimensional static arrays.
- 3210 24. Demonstrate and show implementation of dynamic arrays.

- 3210 25. Understanding of the differences between static and dynamic arrays.
- 3210 26. Knowledge of selection, insertion, merge, and quick sort.
- 3210 27. Demonstrate and show implementation of compiler directives.

## 10159 IB Computing

*IB Computer Studies courses prepare students to take the International Baccalaureate Computing Studies exam at either the Subsidiary or Higher level. The courses emphasize problem analysis, efficient use of data structures and manipulation procedures, and logical decision-making. IB Computing Studies courses also cover the applications and effects of the computer on modern society as well as the limitations of computer technology.*

- 3210 1. Computing System Fundamentals
  - a. Learning Java - basics: data members, applets, simple data types, simple constructs, programming style.
  - b. Sequence, selection and repetition in depth.
  - c. Methods and parameters
- 3 2 1 0 2. Networks
  - a. Mini project - usability issues
- 3 2 1 0 3. Computer Systems
  - a. Classes
  - b. Applications,
  - c. OOP/Design/Programming Concepts
  - d. Testing solutions
  - e. File Management
- 3 2 1 0 4. System Life Cycle
  - a. Preparation for dossier project
- 3 2 1 0 5. Algorithms
  - a. Dossier work
- 32 1 0 6. The Case Study
  - a. Dossier work
- 3 2 1 0 7. Algorithm practice with JETS

## 10160 Particular Topics in Computer Programming (e.g. workplace experience, game design, authoring the web)

*Coursework should represent explicit objectives measured against specific target employment skills that are not available in other courses and should be enumerated in addition to those listed below. Possible topics (you will have others): Data Structures Object Characteristics & Methods of Advanced Algorithms*

*Additional competencies should reflect the particular work environment and the essential skills addressed reflective of previous coursework.*

- 3 2 1 0 1. Summarize the process of IT product/service design.
- 3 2 1 0 2. Plan for products/services using reliability factors.
- 3 2 1 0 3. Create products/services using reliability factors
- 3 2 1 0 4. Test new products/services for reliability
- 3 2 1 0 5. Maintain the reliability of new products/services.

- 3 2 1 0 6. Identify input and output requirements,
- 3 2 1 0 7. Identify system processing requirements.
- 3 2 1 0 8. Define scope of work to meet customer needs.
- 3 2 1 0 9. Demonstrate knowledge of the key functions and subsystems of the software product.
- 3 2 1 0 10. Demonstrate knowledge of cross- functional team structures and team members' roles.
- 3 2 1 0 11. Assess the importance of new technology to future developments.
- 3 2 1 0 12. Identify data communication trends and major current issues.
- 3 2 1 0 13. Identify new technologies relevant to information technology.
- 3 2 1 0 14. Identify system processing requirements.
- 3 2 1 0 15. Determine compatibility of hardware and software.
- 3 2 1 0 16. Identify new and emerging classes of software.
- 3 2 1 0 17. Identify the elements of the information processing cycle (i.e., input, process, output, storage)
- 3 2 1 0 18. Demonstrate knowledge of software development environment.
- 3 2 1 0 19. Develop programs using appropriate language.
- 3 2 1 0 20. Demonstrate knowledge of the information system life cycle.
- 3 2 1 0 21. Demonstrate knowledge of the concepts of data and procedural representations.
- 3 2 1 0 22. Demonstrate knowledge of key constructs and commands specific to a language.
- 3 2 1 0 23. Demonstrate knowledge of how programming control structures are used to verify correctness

### **31098 Programming and Software Development Project Management IT Specific Competencies**

- 3210 1. Summarize the process of IT product/service design.
- 3210 2. Plan for products/services using reliability factors.
- 3210 3. Create products/services using reliability factors
- 3210 4. Test new products/services for reliability
- 3210 5. Maintain the reliability of new products/services.
- 3210 6. Identify input and output requirements,
- 3210 7. Identify system processing requirements.
- 3210 8. Define scope of work to meet customer needs.
- 3210 9. Demonstrate knowledge of the key functions and subsystems of the software product.
- 3210 10. Demonstrate knowledge of cross- functional team structures and team members' roles.
- 3210 11. Assess the importance of new technology to future developments.
- 3210 12. Identify data communication trends and major current issues.
- 3210 13. Identify new technologies relevant to information technology.
- 3210 14. Identify system processing requirements.
- 3210 15. Determine compatibility of hardware and software.
- 3210 16. Identify new and emerging classes of software.
- 3210 17. Identify the elements of the information processing cycle (i.e., input, process, output, storage)
- 3210 18. Demonstrate knowledge of software development environment.
- 3210 19. Develop programs using appropriate language.
- 3210 20. Demonstrate knowledge of the information system life cycle.
- 3210 21. Demonstrate knowledge of the concepts of data and procedural representations.
- 3210 22. Demonstrate knowledge of key constructs and commands specific to a language.
- 3210 23. Demonstrate knowledge of how programming control structures are used to verify correctness

### **Project Management Specific Competencies**

- 3 2 1 0 1. Recognize different resource types (Work, Material, Cost, Budget, Personnel/Skills, Generic, etc)
- 3 2 1 0 2. Understand the concept of scope and demonstrate in context of assessing the size of a project.
- 3 2 1 0 3. Develop plans for project management and resource scheduling.
- 3 2 1 0 4. Identify key personnel and responsibilities for project.
- 3 2 1 0 5. Develop SWOT analysis [Strengths, Weaknesses, Opportunities, and Threats] for project.
- 3 2 1 0 6. Analyze workload of tasks and projects.
- 3 2 1 0 7. Determine required personnel groups and management hierarchy.
- 3 2 1 0 8. Determine resources necessary for project completion.
- 3 2 1 0 9. Determine essential tasks necessary for project completion.
- 3 2 1 0 10. Design potential timelines for assignments.
- 3 2 1 0 11. Explore appropriate technologies for project management and resource scheduling.
- 3 2 1 0 12. Create and present a project management and resource scheduling plan.
- 3 2 1 0 13. Create Gantt charts.
- 3 2 1 0 14. Evaluate and assign resources to tasks.
- 3 2 1 0 15. Implement project management skills to design and complete a collaborative project.
- 3 2 1 0 16. Learn various survey strategies to track project progress.
- 3 2 1 0 17. Develop strategies for monitoring interconnected assignments.
- 3 2 1 0 18. Survey strategies for critical path scheduling.
- 3 2 1 0 19. Create strategies to manage project budgets.
- 3 2 1 0 20. Build survey analysis for customer satisfaction.