CompuScholar, Inc.

Alignment to Ohio's "145020 Computer and Mobile Applications" Standards

Course Title: **TeenCoder: Android Programming**

Course ISBN: **978-0-9887070-5-4**

Course Year: **2015**

Grades: **9th - 12th grade** (high school)

Career Field	Information Technology
Course Name	145020 - Computer and Mobile Applications
Description	Students will learn to create applications for mobile devices using a variety of commercial and open source software. They will install these applications, modify them, and develop customer service skills to handle user issues. Knowledge and skills related to customer service in professional offices, small businesses, departments, work groups, and corporate information services will be addressed.

Note 1: The web design and digital media topics listed in this standard are not taught in Mobile App Development Those topics are covered in our **KidCoder: Web Design** course and marked as "**see KCWD**".

Strand	2. IT Fundamentals	
Description	Learners apply fundamental principles of IT, including the history of IT and its impact on society, common industry terms, systems theory, information storage and retrieval, database management, and computer hardware, software, and peripheral device configuration and installation. This base of knowledge and skills may be applied across the career field.	
Outcome	**2.9. Project Concept Proposal: Develop a project concept proposal.**	**CITATION(S)**
2.9.1. Identify and incorporate branding strategies.		n/a
2.9.2. Determine the scope and purpose of the project.		n/a
2.9.3. Determine the target audience, client needs, expected outcomes, objectives, and budget.		n/a
2.9.4. Develop a conceptual model and design brief for the project.		n/a
2.9.5. Develop a timeline, communication plan, task breakdown, costs (e.g., equipment, labor), deliverables, and responsibilities for completion.		n/a
2.9.6. Develop and present a comprehensive proposal to stakeholders.		n/a

Outcome	2.11. Troubleshooting: Select and apply troubleshooting methodologies for problem solving.	CITATION(S)
2.11.1. Identify the problem.		Chapters 8 and 23
2.11.2. Select troubleshooting methodology (e.g., top down, bottom up, follow the path, spot the differences).		Chapters 8 and 23
2.11.3. Investigate symptoms based on the selected methodology.		Chapters 8 and 23
2.11.4. Gather and analyze data about the problem.		Chapters 8 and 23
2.11.5. Design a solution.		Chapters 8 and 23
2.11.6. Test a solution.		Chapters 8 and 23
2.11.7. Implement a solution.		Chapters 8 and 23
2.11.8. Document the problem and the verified solution.		Chapters 8 and 23
Outcome	**2.12. Performance Tests and Acceptance Plans: Develop performance tests and acceptance plans.**	**CITATION(S)**
2.12.1. Create a written procedure agreed by the stakeholders and project team for determining the acceptability of the project deliverables.		n/a
2.12.2. Develop a test system that accurately mimics external interfaces.		n/a
2.12.3. Develop test cases that are realistic, compare with expected performance, and include targeted platforms and device types.		n/a
2.12.4. Develop, perform, and document usability and testing integration.		n/a
2.12.5. Make corrections indicated by test results.		n/a
2.12.6. Seek stakeholder acceptance upon successful completion of the test plan.		n/a
Outcome	**2.13. Rollout and Handoff: Plan rollout and facilitate handoff to customer.**	**CITATION(S)**
2.13.1. Include overall project goals and timelines in the rollout plan.		n/a
2.13.2. Communicate rollout plans to key stakeholders in a timely manner.		n/a
2.13.3. Conduct final review and approvals according to company standards.		n/a
2.13.4. Identify support staff, training needs, and contingency plans in the rollout plan.		n/a
2.13.5. Test delivered application to assure that it is fully functional for the customer or user and meets all requirements.		n/a

2.13.6. Deliver support and training materials.	n/a

Strand	3. Information Security	
Description	Learners apply principles of information security to implement and maintain security compliance and network security. Learners select components and mechanisms required for a multilayer defense structure and evaluate and minimize security risks to wired and wireless networks and devices.	
Outcome	**3.2. General Security Compliance: Implement and maintain general security compliance.**	**CITATION(S)**
3.2.1. Identify and implement data and application security.		n/a
3.2.2. Implement backup and verification procedures (e.g., tape, disk, cloud).		n/a
3.2.3. Describe and assign permissions (e.g., read-only, read-write).		n/a
3.2.4. Provide user authentication (e.g., assign and reset user accounts and passwords).		n/a
3.2.5. Install, test, implement, and update virus and malware detection and protection software.		n/a
3.2.6. Identify sources of virus and malware infection and remove viruses and malware.		n/a
3.2.7. Provide documentation, training, and support to users on established security procedures.		n/a
3.2.8. Identify the need for disaster recovery policies and procedures.		n/a

Strand	5. Programming and Software Systems	
Description	Learners apply principles of computer programming and software development to develop code; build, test, and debug programs; create finished products; and plan, analyze, design, develop, implement, and support software applications.	
Outcome	**5.1. Programming Concepts: Describe programming concepts.**	**CITATION(S)**
5.1.1. Describe how computer programs and scripts can be used to solve problems (e.g., desktop, mobile, enterprise).		Chapter 1, Lesson 1 Chapter 16, Lesson 3
5.1.2. Explain how algorithms and data structures are used in information processing.		Chapter 3, Lessons 1 - 2 Chapter 9, Lesson 2 Chapter 13, Lessons 1 - 2
5.1.3. Model the solution using both graphic tools (e.g., flowcharts) and pseudocode techniques.		Most activities include instructions analogous to pseudocode

<table>
<tr><td colspan="2">5.1.4. Describe, compare, and contrast the basics of procedural, structured, object-oriented (OO), and event-driven programming.</td><td>Chapter 1, Lesson 2
Chapter 7, Lesson 1
Chapter 9, Lesson 1
Chapter 11, Lesson 3</td></tr>
<tr><td colspan="2">5.1.5. Describe the concepts of data management through programming languages.</td><td>Chapter 3, Lessons 1 - 2
Chapter 9, Lesson 2
Chapter 13, Lessons 1 - 2</td></tr>
<tr><td colspan="2">5.1.6. Analyze the strengths and weaknesses of different languages for solving a specific problem.</td><td>n/a</td></tr>
<tr><td colspan="2">5.1.7. Compare and contrast the functions and operations of compilers and interpreters.</td><td>Chapter 1, Lesson 1</td></tr>
<tr><td colspan="2">5.1.8. Describe version control and the relevance of documentation.</td><td>n/a</td></tr>
<tr><td><b>Outcome</b></td><td><b>5.2. Computational and String Operations: Develop code that performs computational and string operations.</b></td><td><b>CITATION(S)</b></td></tr>
<tr><td colspan="2">5.2.1. Compare and contrast primitive types of numeric and nonnumeric data (e.g., integers, floats, Boolean, strings).</td><td>Chapter 3, Lessons 1 - 2
Chapter 4, Lesson 1</td></tr>
<tr><td colspan="2">5.2.2. Identify the scope of data (e.g., global versus local, variables, constants, arrays).</td><td>Chapter 9, Lesson 2</td></tr>
<tr><td colspan="2">5.2.3. Write code that uses arithmetic operations.</td><td>Chapter 3, Lesson 2 and throughout the course</td></tr>
<tr><td colspan="2">5.2.4. Write code that uses subtotals and final totals.</td><td>Throughout the course</td></tr>
<tr><td colspan="2">5.2.5. Write code that applies string operations (e.g., concatenation, pattern matching, substring).</td><td>Chapter 4, Lessons 1 - 5</td></tr>
<tr><td><b>Outcome</b></td><td><b>5.3. Logical Operations and Control Structures: Develop code that uses logical operations and control structures.</b></td><td><b>CITATION(S)</b></td></tr>
<tr><td colspan="2">5.3.1. Explain Boolean logic.</td><td>Chapter 6, Lesson 1</td></tr>
<tr><td colspan="2">5.3.2. Solve a truth table.</td><td>n/a</td></tr>
<tr><td colspan="2">5.3.3. Write code that uses logical operators (e.g., and, or, not).</td><td>Chapter 6, Lesson 1 and throughout the course</td></tr>
<tr><td colspan="2">5.3.4. Write code that uses relational operators and compound conditions.</td><td>Chapter 6, Lesson 1 and throughout the course</td></tr>
<tr><td colspan="2">5.3.5. Write code that uses conditional control structures (e.g. if, if-then-else).</td><td>Chapter 6, Lesson 2 and throughout the course</td></tr>
<tr><td colspan="2">5.3.6. Write code that uses repetition control structures (e.g., while, for).</td><td>Chapter 6, Lesson 4 and throughout the course</td></tr>
<tr><td colspan="2">5.3.7. Write code that uses selection control structures (e.g., case, switch).</td><td>Chapter 6, Lesson 3 and throughout the course</td></tr>
<tr><td colspan="2">5.3.8. Write code that uses nested structures and recursion.</td><td>n/a</td></tr>
</table>

5.3.9. Write code that creates and calls functions.		Chapter 7, Lessons 1 - 3 and subsequent activities
5.3.10. Code error-handling techniques.		Chapter 5, Lesson 3 Chapter 8, Lesson 2
5.3.11. Write code to access data repositories.		Chapter 18, Lessons 2 - 3
5.3.12. Write code to create classes, objects, and methods.		Chapters 9, 10, 14, 15
Outcome	**5.5. Programming Conventions: Develop programs using applications security best practices according to information security policies (e.g., cross- site scripting, Structured Query Language [SQL] injection attack, bounds- checking).**	**CITATION(S)**
5.5.1. Develop programs using data validation techniques.		Chapter 5, Lesson 3
5.5.2. Develop programs that use reuse libraries.		Android SDK used in Chapters 17+
5.5.3. Develop programs using operating system calls.		JRE and Java class libraries used throughout the course
5.5.4. Develop programs that call other programs.		Chapter 26, Lessons 1 - 2
5.5.5. Use appropriate naming conventions and apply comments.		Chapter 1, Lesson 2 Chapter 3, Lesson 2
5.5.6. Format output (e.g., desktop, mobile, enterprise, reports, data files).		Chapter 4, Lessons 3 - 5 Chapters 20, 21
Outcome	**5.6. Software Development Lifecycle: Apply the software development lifecycle (SDLC).**	**CITATION(S)**
5.6.1. Determine requirements specification documentation.		n/a
5.6.2. Identify constraints and system processing requirements.		n/a
5.6.3. Develop and adhere to timelines.		n/a
5.6.4. Identify a programming language, framework, and an integrated development environment (IDE).		Chapter 1, Lesson 1 Chapter 2, Lesson 1 Chapter 16, Lesson 3
5.6.5. Identify input and output (I/O) requirements.		n/a
5.6.6. Design system inputs, outputs, and processes.		n/a
5.6.7. Document a design using the appropriate tools (e.g., program flowchart, dataflow diagrams, Unified Modeling Language [UML]).		n/a

5.6.8. Create documentation (e.g., implementation plan, contingency plan, data dictionary, user help).		n/a
5.6.9. Review the design (e.g., peer walkthrough).		n/a
5.6.10. Present system design to stakeholders.		n/a
5.6.11. Develop the application.		n/a
5.6.12. Compare and contrast software methodologies (e.g., agile, waterfall).		n/a
5.6.13. Perform code reviews (e.g., peer walkthrough, static analysis).		n/a
5.6.14. Ensure code quality by testing and debugging the application (e.g., system testing, user acceptance testing).		Chapter 8
5.6.15. Train stakeholders.		n/a
5.6.16. Deploy the application.		n/a
5.6.17. Collect application feedback and maintain the application.		n/a
Outcome	**5.7. Configuration Management: Describe configuration management activities.**	**CITATION(S)**
5.7.1. Explain version management and interface control.		n/a
5.7.2. Explain baseline and software lifecycle phases.		n/a
5.7.3. Analyze the impact of changes.		n/a

Strand	6. Web Development	
Description	Learners apply principles of design and technology, including programming standards and protocols, to create, test, host, and maintain webpages and websites with text, graphics, multimedia, scripting, linking, and data integration in a structure that is easy to navigate and accessible for all users via a variety of hardware and software platforms.	
Outcome	**6.2. Links and Multimedia: Add links to a webpage and insert multimedia files.**	**CITATION(S)**
6.2.1. Create absolute links and relative links.		see KCWD
6.2.2. Write a Hypertext Markup Language (HTML) anchor that links to another section of the same webpage.		see KCWD
6.2.3. Create hyperlinks that send e-mail messages and download files.		see KCWD
6.2.4. Insert image and wrap text around the image using Cascading Style Sheets (CSS).		see KCWD

6.2.5. Resize a graphic image in a webpage using CSS.		see KCWD
6.2.6. Insert audio and video files into a webpage using HTML tags.		see KCWD
6.2.7. Build a hover or mouseover effect to change the style of a link.		see KCWD
Outcome	**6.3. Scripting: Integrate scripting into a webpage.**	**CITATION(S)**
6.3.1. Select and apply scripting languages used in web development.		see KCWD
6.3.2. Insert client-side script into a webpage.		see KCWD
6.3.3. Insert comments into client-side scripts.		see KCWD

Strand	7. Digital Media	
Description	Learners apply principles of digital media to produce interactive media; develop and produce multimedia applications; integrate typography into media; create 3D models and 2D and 3D animation; and create digital video, audio, and photographs.	
Outcome	**7.2. Multimedia Tools: Develop navigational structures, scripts, storyboards, and flowcharts for multimedia applications.**	**CITATION(S)**
7.2.1. Choose a navigational menu structure (e.g., rollovers, drop-downs, disjointed).		see KCWD
7.2.2. Construct and place navigational units.		see KCWD
7.2.3. Build in interactive elements.		Interactive programs and games are developed throughout the course
7.2.4. Determine uses and needs for site maps, multimedia scripts, storyboards, and flowcharts.		see KCWD
7.2.5. Make preliminary sketches showing placement of images and text on screen.		see KCWD
7.2.6. Show placement of buttons and navigational graphics.		see KCWD
7.2.7. Provide information on color schemes.		see KCWD
7.2.8. Describe music, video, and special effects to be used.		see KCWD
7.2.9. Provide a sample layout to stakeholders for review.		see KCWD
7.2.10. Select and create visual design elements appropriate for the intended audience and use.		see KCWD

7.2.11. Develop characters and narrative to support intended outcomes.		see KCWD
Outcome	**7.4. Graphics: Construct and manipulate digital graphics.**	**CITATION(S)**
7.4.1. Identify the purpose and intended audience of graphics.		see KCWD
7.4.2. Select color, shape, size, and texture of objects.		see KCWD
7.4.3. Create or acquire graphics.		see KCWD
7.4.4. Manipulate and layer objects.		see KCWD
7.4.5. Differentiate between vector and raster images.		see KCWD
7.4.6. Select an appropriate graphic file format and resolution.		Chapter 24, Lesson 1
7.4.7. Optimize and export graphics files for intended use.		see KCWD
7.4.8. Select graphic software applications.		see KCWD
7.4.9. Manipulate graphic objects.		see KCWD
7.4.10. Compress and decompress graphic files.		see KCWD
7.4.11. Describe and select color profiles (e.g., Red Green Blue [RGB], Cyan Magenta Yellow Key [CMYK], Pantone)		see KCWD