Grade 8 BCIT Course: Business, Computer & Information Technologies
Unit 1: Course Introduction

Enduring Understandings:

- Proficiency in computers is an absolute requirement for success in today's digital, interconnected, and rapidly evolving world.

Essential Questions:

- What skills will I learn in this class and how can I apply them now and in the future?
- What technology and computer science skills are the most in demand?

Content ***What will be taught?***	***Objectives*** ***What will students know & be able to do as a result of this instruction?***	***Area of Focus/ Instructional Activities/ Lessons*** ***What will students do to achieve the objectives?***	***Options for Modifications/Extensions*** ***How will curriculum be differentiated to meet individual student needs?***	***Assessments*** ***What evidence will be collected to demonstrate students have achieved the objectives?***	***Resources*** ***What materials will be used to achieve the objectives?***
Class Overview & Expectations	Summarize and explain: • Course content • Classroom expectations • Neshaminy's Acceptable Use Policy (AUP) Adhere to responsible use guidelines when using technology. Access Neshaminy's network. Research and evaluate job growth, career outlook, and other statistics on information technology and computer science careers.	Read, summarize, and discuss classroom goals, expectations, and essential excerpts of Neshaminy's AUP. Identify behaviors that constitute misuse of technology. Abide by Neshaminy's AUP on a daily basis. Independently log in to Neshaminy's network using the District assigned username and password. Research IT and CS occupations identifying: • Career options • Skills and education needed • Supply & demand of CS, IT workers • Salary • Future job outlook Link individual interests & aptitudes to those required for IT/CS careers.	Course information will be posted on teacher website or other online service. Teacher assistance with forgotten username or password. Data on IT/CS careers provided to students, per IEP or 504 Plans. Clinic and/or WIN period assistance given as necessary.	Participation and feedback from class discussions.	Neshaminy's Acceptable Use Policy http://www.neshaminy.org/cms/lib6/PA01000466/Centricity/Domain/7/bd%20pol%20811.pdf School's student login credentials list. Occupational Outlook Handbook (Computer and Information Technology Occupations) https://www.bls.gov/ooh/computer-and-information-technology/home.htm Teacher-selected videos

Content ***What will be taught?***	***Objectives*** ***What will students know & be able to do as a result of this instruction?***	***Area of Focus/ Instructional Activities/ Lessons*** ***What will students do to achieve the objectives?***	***Options for Modifications/Extensions*** ***How will curriculum be differentiated to meet individual student needs?***	***Assessments*** ***What evidence will be collected to demonstrate students have achieved the objectives?***	***Resources*** ***What materials will be used to achieve the objectives?***

Basic Computer Troubleshooting	Troubleshoot and correct basic computer problems and login issues.	Use troubleshooting techniques to identify and correct basic computer problems and login issues:
		• Check system unit for power
		• Check monitor for power
		• Verify power to keyboard and mouse
		• Verify the URL
		• Verify network connectivity
		• Close and reopen the program
		• Try on a different web browser
		• Login issues:
		  ○ Forgotten username or password
		  ○ Caps lock
		  ○ Num lock
		• Report urgent issues to the teacher immediately. Leave computer untouched:
		  ○ Pop-up messages or virus warnings
		  ○ Strange activity
		  ○ Music playing in the background, even when logged out
		
		Properly log off when leaving the computer (WIN + L if you need a quick log out).
			Teacher-created handouts
			
			Teacher assistance
				Anecdotal evidence	Online videos and teacher demonstrations of basic troubleshooting
					
					School's list of student usernames and passwords

Google Chrome Bookmarking	Increase efficiency by using bookmarks in Google Chrome.	Navigate to frequented websites, creating bookmarks for each one in Google Chrome.	Clinic and/or WIN period assistance	Observation of bookmarks.	

Standards: ***BCIT:*** *15.2.8.B, 15.2.8.C, 15.2.8.D, 15.2.8.E, 15.2.8.G, 15.2.8.L, 15.2.8.M, 15.2.8.O, 15.2.8.P, 15.2.8.Q, 15.3.8.C, 15.3.8.D, 15.3.8.F, 15.3.8.G, 15.3.8.H, 15.3.8.I, 15.3.8.J, 15.3.8.K, 15.3.8.L, 15.3.8.M, 15.3.8.N, 15.3.8.S, 15.3.8.T, 15.3.8.X, 15.4.8.A, 15.4.8.B, 15.4.8.D, 15.4.8.F, 15.4.8.M, 15.6.8.M, 15.7.8.I, 15.8.8.C, 15.8.8.I, 15.8.8.K* ***Career Ed. & Work:*** *13.1.8.A, 13.1.8.B, 13.1.8.D, 13.1.8.E, 13.2.8.B, 13.2.8.E* ***ISTE-NETS****: 1A, 1B, 1C, 1D, 2A, 2B, 2C, 2D, 3A, 3B, 3C, 3D, 4A, 4B, 4C, 4D, 5A, 5B, 5C, 5D, 6A, 6B, 6C, 6D, 7B, 7C* ***CSTA:*** *2-C-6-13, 2-D-7-15, 2-I-1-21, 2-I-1-22, 2-I-7-19, 2-N-7-24* ***CC-ELA:*** *CC.1.2.8.A, CC.1.2.8.B, CC.1.2.8.C, CC.1.2.8.H, CC.1.2.8.I, CC.1.2.8.J, CC.1.2.8.K, CC.1.4.8.F, CC.1.4.8.H, CC.1.5.8.A, CC.1.5.8.B, CC.1.5.8.E, CC.1.5.8.G* ***CC-Reading in Science & Tech.:*** *CC.3.5.6-8.C, CC.3.5.6-8.D, CC.3.5.6-8.G, CC.3.5.6-8.J* ***CC-Writing in Science & Tech.:*** *CC.3.6.6-8.C, CC.3.6.6-8.D, CC.3.6.6-8.E, CC.3.6.6-8.H*

Grade 8 BCIT Course: Business, Computer & Information Technologies

Unit 2: Alice 3D Programming with Java

Enduring Understandings:

- Alice is an object-oriented programming language based on Java programming.
- Alice programming provides a new way of thinking and helps develop perseverance and problem-solving skills.

Essential Questions:

- How do I get the computer to do what I want it to do in Alice?
- How do I use classes, object, procedures, and methods to create a virtual world in Alice?

Content ***What will be taught?***	***Objectives*** ***What will students know & be able to do as a result of this instruction?***	***Area of Focus/ Instructional Activities/ Lessons*** ***What will students do to achieve the objectives?***	***Options for Modifications/Extensions*** ***How will curriculum be differentiated to meet individual student needs?***	***Assessments*** ***What evidence will be collected to demonstrate students have achieved the objectives?***	***Resources*** ***What materials will be used to achieve the objectives?***
Alice 3D Programming	Identify the link between Alice and Java programming. Formulate object-oriented programming statements in Alice to create, design, and customize 3D animations and games. Manipulate objects and apply procedures using the Scene Editor and the Code Editor. Use the iterative design process to test and revise programming statements as part of the debugging process.	View Alice projects comparing its features to those in Scratch (learned in 7th grade BCIT classes). Review career opportunities and lack of diversity in computer science careers. (see Unit 1) Recall and incorporate proper Alice terminology into communications. (see Appendix G, H, I, J, & K) Differentiate the Scene Editor and Code Editor within the Alice interface. Assemble multiple programming sequences in Alice to create virtual 3D simulations of real-life and fictional scenarios. Research textbooks, online sources, and printed materials for solutions to programming issues.	Collaborate on projects with a partner or work alone. Debugging warm-ups created by teacher. Insert audio resources into projects: background music, musical cues, and sounds. Self-paced projects with step-by-step directions provided to students with IEPs or 504 Plans. Add more than one scene to a project. Create a project with billboards. View Java code beside the Alice code.	Teacher-created assignments and tests: • Alice vocab. • Alice interface • Alice projects Periodic check-ins with teacher	Getting Started with Java using Alice 3 http://www.oracle.com/webfolder/technetwork/tutorials/OracleAcademy/Alice3SelfStudyV2/index.html#overview Alice 3 Course Materials: http://www.alice.org/3.1/index.html http://www.cs.duke.edu/csed/alice09/tutorialsAlice3.php Assessments, Homework, Videos, and Lesson Plans—Resource Library http://www.curriki.org/oer/getting-started-with-java-using-alice-59889/ https://www.opened.com/

Content ***What will be taught?***	***Objectives*** ***What will students know & be able to do as a result of this instruction?***	***Area of Focus/ Instructional Activities/ Lessons*** ***What will students do to achieve the objectives?***	***Options for Modifications/Extensions*** ***How will curriculum be differentiated to meet individual student needs?***	***Assessments*** ***What evidence will be collected to demonstrate students have achieved the objectives?***	***Resources*** ***What materials will be used to achieve the objectives?***
		Add objects to a scene: resize, position, and change an object's properties and orientation. Distinguish among the Handle Styles to edit, move, and resize objects. Position objects using precise positioning, one-shot procedures, and the drag-and-drop technique. Position, rotate, and animate the sub-parts (internal joints) of objects. Set the camera's vehicle to moving objects. Declare user-defined procedures, such as walking and talking. Define and use multiple control statements to control animation timing, duration, and other values. Reorder, edit, delete, copy, and disable programming statements. Identify and correct syntax errors, logic errors, and run time errors. Use the iterative design process throughout the creation of projects: prototype, test, revise, and refine.	Add Java programming procedures to the Code Editor. #iSTEAMM trip		

Standards: ***BCIT:*** *15.2.8.C, 15.2.8.D, 15.2.8.E, 15.2.8.G, 15.2.8.L, 15.2.8.M, 15.2.8.O, 15.2.8.P, 15.2.8.Q, 15.3.8.C, 15.3.8.D, 15.3.8.E, 15.3.8.F, 15.3.8.G, 15.3.8.H, 15.3.8.I, 15.3.8.J, 15.3.8.K, 15.3.8.L, 15.3.8.M, 15.3.8.N, 15.3.8.S, 15.3.8.T, 15.3.8.U, 15.3.8.W, 15.3.8.X, 15.4.8.A, 15.4.8.B, 15.4.8.D, 15.4.8.F, 15.4.8.G, 15.4.8.H, 15.4.8.I, 15.4.8.J, 15.4.8.K, 15.4.8.M, 15.8.8.I, 15.8.8.J, 15.8.8.K* ***ISTE-NETS:*** *1A, 1B, 1C, 1D, 2A, 2B, 2C, 2D, 3A, 3B, 3C, 3D, 4A, 4B, 4C, 4D, 5A, 5B, 5C, 5D, 6A, 6B, 6C, 6D, 7B, 7C* ***CSTA:*** *2-A-2-1, 2-A-3-9, 2-A-4-8, 2-A-5-5, 2-A-5-6, 2-A-6-10, 2-A-7-2, 2-A-7-3, 2-A-7-4, 2-C-7-11, 2-D-5-16, 2-D-5-17, 2-D-7-15, 2-I-7-19, 2-I-1-21, 2-I-6-23* ***CC-Reading in Science & Tech.:*** *CC.3.5.6-8.C, CC.3.5.6-8.D, CC.3.5.6-8.G, CC.3.5.6-8.J* ***Career Ed. & Work:*** *13.3.8.B* ***Arts & Humanities****: 9.1.8.B, 9.1.8.C, 9.1.8.E* ***Math****: CC.2.2.8.C.1* ***CC-ELA****: CC.1.2.8.J, CC.1.2.8.K, CC.1.2.8.L, CC.1.4.8.F, CC.1.4.8.H, CC.1.4.8.S, CC.1.4.8.U, CC.1.4.8.V, CC.1.5.8.A, CC.1.5.8.B, CC.1.5.8.E, CC.1.5.8.F, CC.1.5.8.G*

Grade 8 BCIT Course: Business, Computer & Information Technologies
Unit 3: Google Drive & Google Docs

Enduring Understandings:

- G Suite is a cloud-based collaborative suite of tools used to create, store, and share files, such as Google Docs, Google Drive, Google Slides, and many more.
- Google Docs is an online word processing software tool allowing accessibility to documents from any device with Internet access.

Essential Questions:

- What is G Suite?
- How can I create and share my Google Docs files with others?

Content ***What will be taught?***	***Objectives*** ***What will students know & be able to do as a result of this instruction?***	***Area of Focus/ Instructional Activities/ Lessons*** ***What will students do to achieve the objectives?***	***Options for Modifications/Extensions*** ***How will curriculum be differentiated to meet individual student needs?***	***Assessments*** ***What evidence will be collected to demonstrate students have achieved the objectives?***	***Resources*** ***What materials will be used to achieve the objectives?***
Google Drive	Explain what G Suite is and what it consists of. Explain the advantages of using Google Drive for storing, sharing, and accessing files. Store and organize files in Google Drive by creating folders for document categories. Set a document's visibility and permissions, sharing files with the teacher and/or peers for collaboration.	Access Google Drive using the District assigned username and password. Utilize Google Drive to create, save, open, share, move, delete, and review documents. Create folders and subfolders within Google Drive to organize files into meaningful categories. Display the list of files in Google Drive as both a grid and a list view. Use Google Drive to upload non-Google files. Participate in classroom discussions and provide feedback relating to Google Drive, cloud-based storage, and document organization.	Clinic and/or WIN period assistance provided, as necessary.	Teacher-guided practice	Intro. to G Suite for Education https://youtu.be/qnAYyMG39sk Using Google Drive Information Web Page https://support.google.com/drive/?hl=en#topic=14940 Teacher-created materials

Content ***What will be taught?***	***Objectives*** ***What will students know & be able to do as a result of this instruction?***	***Area of Focus/ Instructional Activities/ Lessons*** ***What will students do to achieve the objectives?***	***Options for Modifications/Extensions*** ***How will curriculum be differentiated to meet individual student needs?***	***Assessments*** ***What evidence will be collected to demonstrate students have achieved the objectives?***	***Resources*** ***What materials will be used to achieve the objectives?***
Google Docs	Create, save, format, and share documents created in Google Docs. Increase efficiency by using keyboard shortcuts.	Use Google Docs for note taking: • HTML vocabulary • HTML notes/reference • CSS vocabulary • CSS notes/reference • Alice vocabulary • Alice notes/reference Format documents in Google Docs: • Text style, color, size, highlight color, and alignment • Bold, italics, underline • Change text case (lower, UPPER, Title Case) • Paragraph spacing • Undo/redo actions • Page orientation & margins • Keyboard shortcuts • Copy and paste with formatting • Copy and paste without formatting • Insert and format images • Hyperlinks • Spelling/grammar check	Use a template for document creation. Advanced formatting: • Insert/format tables • Create/format lists • Insert/format text boxes • Insert videos Create and format documents in Google Sheets. Create and format documents in Google Slides. Create a survey in Google Forms.		List of Google keyboard shortcuts can be viewed and printed in Google: Press CTRL + /

Standards: ***BCIT:*** 15.2.8.G, 15.2.8.L, 15.2.8.M, 15.3.8.A, 15.3.8.B, 15.3.8.C, 15.3.8.D, 15.3.8.E, *15.3.8.F, 15.3.8.G, 15.3.8.H, 15.3.8.I, 15.3.8.J, 15.3.8.K, 15.3.8.L, 15.3.8.M, 15.3.8.N, 15.3.8.S, 15.3.8.T, 15.3.8.U, 15.3.8.V, 15.3.8.W, 15.3.8.X, 15.4.8.A, 15.4.8.B, 15.4.8.D, 15.4.8.F, 15.4.8.K, 15.8.8.I* ***ISTE-NETS:*** *2A, 2B, 2C, 2D, 6A, 6B* **CSTA:** *2-I-1-21, 2-I-1-22, 2-I-7-19* ***CC-Reading in Science & Tech.****: CC.3.5.6-8.C, CC.3.5.6-8.G, CC.3.5.6-8.J* ***CC-Writing in Science & Tech.****: CC.3.6.6-8.B, CC.3.6.6-8.C, CC.3.6.6-8.E, CC.3.6.6-8.H, CC.3.6.6-8-I, CC.3.6.6-8.J* ***CC-ELA:*** *CC.1.2.8.J, CC.1.2.8.K, CC.1.2.8.L, CC.1.4.8.A, CC.1.4.8.C, CC.1.4.8.D, CC.1.4.8.E, CC.1.4.8.H, CC.1.5.8.A, CC.1.5.8.B, CC.1.5.8.E, CC.1.5.8.F, CC.1.5.8.G*

Grade 8 BCIT Course: Business, Computer & Information Technologies

Unit 4: HTML & Web Page Design

Enduring Understandings:

- HTML is the standard markup language used to create ALL web pages.
- HTML is used to add the content to web pages while CSS coding is added to apply the formatting and styling of web pages.

Essential Questions:

- What can I do with my knowledge of HTML?
- How do HTML and CSS work together in web page design?

Content ***What will be taught?***	***Objectives*** ***What will students know & be able to do as a result of this instruction?***	***Area of Focus/ Instructional Activities/ Lessons*** ***What will students do to achieve the objectives?***	***Options for Modifications/Extensions*** ***How will curriculum be differentiated to meet individual student needs?***	***Assessments*** ***What evidence will be collected to demonstrate students have achieved the objectives?***	***Resources*** ***What materials will be used to achieve the objectives?***
HTML & CSS ***Web Page Design*** **Notepad++** ***Text Editor*** **Online Text Editor Applications**	Explain what HTML is, what it is used for, and how knowledge of HTML can be useful in almost all careers. Explain what CSS is and how it is used in conjunction with HTML to build web pages. Analyze the source code of various web pages. Build and customize web pages with HTML and CSS. Recall and use proper HTML and CSS terminology. Test and debug web pages. Recognize the proper syntax of HTML.	Research careers in web page design and related computer science fields: • Career options • Education/training, aptitudes, and abilities necessary • Salary and job outlook • Supply vs. demand for CS workers • Lack of diversity within CS fields, especially women and minorities • Methods being used to equalize career opportunities in CS • Complete a Text Dependent Analysis on findings. Interpret and edit the source code of HTML web pages identifying the resulting effects of those coding changes. Recognize and apply the strict and precise syntax rules of HTML and CSS coding. Use Notepad++ as the text editor for HTML and CSS coding.	Teacher-created debugging warm-ups of varying degrees of rigor Bebras Computing Challenge PicoCTF Hacking Competition #iSTEAMM trip Apply different bullet types to unordered lists. Apply different numbering styles to ordered lists. Compare the different methods used to size text, including: pixel, em, & percentage.	Teacher-created assignments Teacher-created assessments: HTML Elements HTML/CSS Vocab. Alice Text Dependent Analysis Student feedback during activities and discussions.	W3 Schools--HTML https://www.w3schools.com/html/ W3 Schools--CSS https://www.w3schools.com/CSS/ TechnoKids Lessons http://www.technokids.com/computer-curriculum/intermediate-computer-curriculum.aspx Bebras Computing Challenge: http://bebraschallenge.org/ TechRepublic—13 Design Rules for Websites http://www.techrepublic.com/article/13-design-rules-that-every-web-site-designer-should-know-about/

Content ***What will be taught?***	***Objectives*** ***What will students know & be able to do as a result of this instruction?***	***Area of Focus/ Instructional Activities/ Lessons*** ***What will students do to achieve the objectives?***	***Options for Modifications/Extensions*** ***How will curriculum be differentiated to meet individual student needs?***	***Assessments*** ***What evidence will be collected to demonstrate students have achieved the objectives?***	***Resources*** ***What materials will be used to achieve the objectives?***
	Recognize the proper syntax of CSS. Apply proper HTML syntax and CSS syntax.	Organize coding statements into an easy-to-follow format by using indents, alignment, spacing, and comments. Recognize and use appropriate HTML and CSS terminology (see Appendix B). Create the foundation and structure of web pages with HTML elements: • Head • Body • Subheadings (h1-h6) • Horizontal rules • Anchors/Hyperlinks • Span • Title • Headings • Paragraphs • Images • Style • Break Use the iterative design process when creating Alice projects: prototype, test, analyze, and refine. Apply appropriate inline CSS statements and internal CSS statements to enhance and style HTML web pages: • Font family, size, and color • Text alignment • Italicize & bold • Line Height • Image alignment • Border style, width, and color • Underline and/or overline • Lists (ordered and unordered) • Color hexadecimal codes • Height & width properties • Padding, borders, and margins	Add and structure additional web page elements such as columns, tables, divisions, rounded borders, beveled borders, classes, & block quotes. Link to an external CSS document. Add drop shadows to elements. Create a gradient background color. Add an interactive JavaScript button such as the current date and time. Compare and contrast: • HTML & CSS • Start & end tags • Ordered & unordered lists • Underlining & overlining		

Content ***What will be taught?***	***Objectives*** ***What will students know & be able to do as a result of this instruction?***	***Area of Focus/ Instructional Activities/ Lessons*** ***What will students do to achieve the objectives?***	***Options for Modifications/Extensions*** ***How will curriculum be differentiated to meet individual student needs?***	***Assessments*** ***What evidence will be collected to demonstrate students have achieved the objectives?***	***Resources*** ***What materials will be used to achieve the objectives?***
Google Sites Web Page Creation	Design a multi-page website in Google Sites, applying HTML and CSS coding. Choose an appropriate topic for a multi-page website, research that topic through several online sources to create valid, informational content for the website. Incorporate various Google Inserts into the website.	Create and design a multi-page website in Google Sites. Change sharing and permissions settings on Google Sites, sharing the website project with a partner (owner) and the teacher (viewer). Design and edit web pages with a partner using Google Site's live collaboration features. Edit the HTML source code, as needed. Recall and use appropriate Google Sites vocabulary (See Appendix F). Identify and format the elements of a web page: • Site header • Content area • Content area gadgets • Sidebar gadgets • Horizontal navigation • System footer Format/style web pages in a professional manner: • Apply themes • Add and delete pages and subpages • Reorder pages • Edit the site name and page titles • Add images, gadgets, and horizontal lines • Create hyperlinks for text and images • Font style, size, color and alignment • Bold, italics, underline • Page design layout • Custom backgrounds	Student option to work with a partner or alone. Website topics will be student-selected based upon their individual interests and skills. **Examples**: *Pets, Pet Care, Visiting Texas, Theme Parks, Covered Bridges of Bucks County, Things to do in PA …* Add advanced elements to Google Sites: • Templates • Add text boxes • Insert videos • Insert table of contents page • Add Layouts • Add Lists	Formative Assessments: periodic check-ins Website project time management Summative Assessments: Google Sites Project	Google Sites Help Center https://support.google.com/sites/?hl=en#topic=7020201 EdSurge https://www.edsurge.com/research/edtech-wiki

Content ***What will be taught?***	***Objectives*** ***What will students know & be able to do as a result of this instruction?***	***Area of Focus/ Instructional Activities/ Lessons*** ***What will students do to achieve the objectives?***	***Options for Modifications/Extensions*** ***How will curriculum be differentiated to meet individual student needs?***	***Assessments*** ***What evidence will be collected to demonstrate students have achieved the objectives?***	***Resources*** ***What materials will be used to achieve the objectives?***
		Employ various troubleshooting methods when creating/debugging web pages: online sources, printed materials, books, tutorials, etc. Assess web pages providing feedback on content, design, and overall professionalism. Participate in classroom discussions using correct CS terminology related to web page design.			

Standards: ***BCIT:*** *15.2.8.C, 15.2.8.D, 15.2.8.E, 15.2.8.G, 15.2.8.L, 15.2.8.M, 15.2.8.O, 15.2.8.P, 15.2.8.Q, 15.3.8.A, 15.3.8.C, 15.3.8.D, 15.3.8.E, 15.3.8.F, 15.3.8.G, 15.3.8.H, 15.3.8.I, 15.3.8.J, 15.3.8.K, 15.3.8.L, 15.3.8.M, 15.3.8.N, 15.3.8.S, 15.3.8.T, 15.3.8.U, 15.3.8.W, 15.3.8.X, 15.4.8.A, 15.4.8.B, 15.4.8.D, 15.4.8.F, 15.4.8.G, 15.4.8.H, 15.4.8.I, 15.4.8.J, 15.4.8.K, 15.4.8.L, 15.4.8.M, 15.8.8.C, 15.8.8.I, 15.8.8.J, 15.8.8.K, 15.8.8.L* ***ISTE-NETS:*** *1A, 1B, 1C, 1D, 2A, 2B, 2C, 2D, 3A, 3B, 3C, 3D, 4A, 4B, 4C, 4D, 5A, 5B, 5C, 5D, 6A, 6B, 6C, 6D, 7B, 7C* ***CSTA:*** *2-A-2-1, 2-A-3-9, 2-A-4-8, 2-A-6-10, 2-A-7-2, 2-A-7-3, 2-A-7-4-, 2-A-5-5, 2-A-5-6, 2-C-7-11, 2-D-7-15, 2-D-5-16, 2-D-5-17, 2-I-1-22, 2-I-1-21, 2-I-6-23, 2-I-7-19* ***CC-Reading in Science & Tech.:*** *CC.3.5.6-8.A, CC.3.5.6-8.B, CC.3.5.6-8.C, CC.3.5.6-8.D, CC.3.5.6-8.E, CC.3.5.6-8.G, CC.3.5.6-8.H, CC.3.5.6-8.J* ***Arts & Humanities:*** *9.1.8.B, 9.1.8.C, 9.1.8.E* ***CC-ELA:*** *CC.1.2.8.A, CC.1.2.8.B, CC.1.2.8.C, CC.1.2.8.E, CC.1.2.8.F, CC.1.2.8.J, CC.1.2.8.K, CC.1.2.8.L, CC.1.4.8.A, CC.1.4.8.B, CC.1.4.8.C, CC.1.4.8.D, CC.1.4.8.E, CC.1.4.8.F, CC.1.4.8.H, CC.1.4.8.S, CC.1.4.8.T, CC.1.4.8.U, CC.1.4.8.V, CC.1.4.8.W, CC.1.4.8.X, CC.1.5.8.A, CC.1.5.8.B, CC.1.5.8.D, CC.1.5.8.E, CC.1.5.8.F, CC.1.5.8.G* ***CC-Writing in Science & Tech.:*** *CC.3.6.6-8.B, CC.3.6.6-8.C, CC.3.6.6-8.D, CC.3.6.6-8.E, CC.3.6.6-8.F, CC.3.6.6-8.H, CC.3.6.6-8.I, CC.3.6.6-8.J*

Grade 8 BCIT Course: Business, Computer & Information Technologies
Unit 5: Financial Literacy & Economics for Success

Enduring Understandings:

- **Awareness of personal skills, interests, and values can guide students in education, career, and life choices.**
- **A budget is a money management tool that helps plan for financial goals and tracks spending.**
- **Paying yourself first (PYF) is a fiscal responsibility that plans for the future.**

Essential Questions:

- **How do education and life choices intersect throughout a lifetime?**
- **What is a budget and why do I need one?**
- **How can I minimize personal financial hardships?**

Content ***What will be taught?***	***Objectives*** ***What will students know & be able to do as a result of this instruction?***	***Area of Focus/ Instructional Activities/ Lessons*** ***What will students do to achieve the objectives?***	***Options for Modifications/Extensions*** ***How will curriculum be differentiated to meet individual student needs?***	***Assessments*** ***What evidence will be collected to demonstrate students have achieved the objectives?***	***Resources*** ***What materials will be used to achieve the objectives?***
Self-Knowledge & Career Choices	Explore personal skills, interests, values, and the world of work to make informed education, career, and life decisions. Develop a knowledge of personal finance to apply strong financial-management skills regardless of income.	Use personal reflection to explain self-knowledge. Explain how self-knowledge contributes to good decision making. Self-assess personal skills, interests, and values to help determine a potential career path. Define career clusters. Explain the relationship between self-knowledge and the working world. Connect with community role models. Participate in classroom discussions.	Research local job listings online, select a job of interest, noting skill requirements of the job. List items needed to apply for the online job and research the company's job-specific website to learn more about the company. Present findings to class.	Online career assessment	Self-knowledge treasure hunt Next Gen Personal Finance Activities Https://www.gpf.org

Content ***What will be taught?***	***Objectives*** ***What will students know & be able to do as a result of this instruction?***	***Area of Focus/ Instructional Activities/ Lessons*** ***What will students do to achieve the objectives?***	***Options for Modifications/Extensions*** ***How will curriculum be differentiated to meet individual student needs?***	***Assessments*** ***What evidence will be collected to demonstrate students have achieved the objectives?***	***Resources*** ***What materials will be used to achieve the objectives?***
Decision Making & Goal Setting	Identify the connections between goal setting, personal finance, education, and career choices. Apply decision making to education and career choices.	Complete the "Spent" challenge Choose an education or work path on the 'Be a Success Game' board and experience a variety of life choices. Explain how education and work choices intersect throughout a lifetime. Describe how setbacks to career goals can be handled. Interact with community role models.	Take a motivation exercise assessment	NextGen Goals Assessment	Be a Success Game http://playspent.org/ Next Gen Personal Finance Activities Https://www.gpf.org
Budgeting	Recognize that a balanced budget is important for all workers. Differentiate between gross and net income.	Create a budget based on sample information provided. Define gross income and net income, differentiating between the two. Name ways to balance a budget. Explain the phrase "pay yourself first." Summarize the relationship between what is learned in school and successful participation in a global economy. Share thoughts and questions with community role models.	Create a chart, putting data in graph form, listing different job choices and the gross monthly income. Compare the earnings of various occupations. Research and report on the specific education and training needed for various jobs.	Budget Assignment Budgeting Basics Assessment-Next Gen	Occupation Card Set Budget Choice Ads Poster Keeping Your Balance Fliers Next Gen Personal Finance Activities Https://www.gpf.org Bizkids http://bizkids.com/

Content ***What will be taught?***	***Objectives*** ***What will students know & be able to do as a result of this instruction?***	***Area of Focus/ Instructional Activities/ Lessons*** ***What will students do to achieve the objectives?***	***Options for Modifications/Extensions*** ***How will curriculum be differentiated to meet individual student needs?***	***Assessments*** ***What evidence will be collected to demonstrate students have achieved the objectives?***	***Resources*** ***What materials will be used to achieve the objectives?***
Credit & Debit	Identify the differences between debit and credit cards. Explain the advantages and disadvantages of debit cards and credit cards. Recognize the importance of taking personal responsibility for financial decisions.	Define credit, credit card, debit card, and interest. List the advantages and the disadvantages of using credit to purchase needs and wants. Make connections with community role models. Participate in classroom discussions.	Complete a chart of best purchasing choices for various items (cash or credit). Calculate the interest and total cost of various items purchased on credit. Utilize the interactive tools on the Consumer Financial Protection Bureau website, learning how to choose and use credit cards.	Formative Assessment: Display of thumbs up/thumbs down in response to credit and banking questions. Credit Basics Assessment-Next Gen	Consumer Financial Protection Bureau website www.consumerfinance.gov Next Gen Personal Finance Activities Https://www.gpf.org
Credit Scores	Explain how a credit score is determined. Identify the favorable or unfavorable consequences of a high or low personal credit score. Explain how a poor credit score limits your choices. Explain the actions that cause a credit score to go up or down.	Define credit score and why a good credit score is important. Identify what constitutes a good credit score and a bad credit score. Make connections between real-life simulations and their effect on credit rating. Observe and evaluate the impact of a positive role model.		Credit Score Assessment-Next Gen	Next Gen Personal Finance Activities Https://www.gpf.org

Standards: ***BCIT:*** *15.1.8.L, 15.1.8.M, 15.1.8.Q, 15.2.8.A, 15.2.8.B, 15.6.8.H, 15.6.8.K, 15.6.8.L, 15.6.8.N* ***ISTE-NETS:*** *1A, 1B, 1C, 1D, 2A, 2B, 2C, 2D, 3A, 3B, 3C, 3D, 4D, 5A, 5B, 5C, 6A, 6B, 6C, 6D, 7B, 7C* ***Career Ed. & Work:*** *13.1.8.A, 13.1.8.B, 13.1.8.D, 13.1.8.E, 13.1.8.F, 13.3.8.D* ***CC-Reading in Science & Tech.****: CC.3.5.6-8.A, CC.3.5.6-8B, CC.3.5.6-8.C, CC.3.5.6-8.D, CC.3.5.6-8.G, CC.3.5.6-8.H, CC.3.5.6-8.I, CC.3.5.6-8.J* ***CC-Math:*** *NS.B.6.3, NS.C.6.5, RP.A.7.E, CC.2.2.8.C.1, CC.2.2.8.C.2, CC.2.3.8.A.2, CC.2.4.8.B,1, CC.2.4.8.B.2* ***CC-Writing in Science & Tech.****: CC.3.6.6-8.A, CC.3.6.6-8.B, CC.3.6.6-8.C, CC.3.6.6-8.E, CC.3.6.6-8.F* ***CC-ELA:*** *CC.1.2.8.A, CC.1.2.8.B, CC.1.2.8.G, CC.1.2.8.H, CC.1.2.8.J, CC.1.2.8.L, CC.1.5.8.A, CC.1.5.8.B, CC.1.5.8.C, CC.1.5.8.D, CC.1.5.8.E, CC.1.5.8.F, CC.1.5.8.G, CC.1.5.8.A, CC.1.5.8.C, CC.1.5.8.G*

Appendix A: HTML Tag Reference List

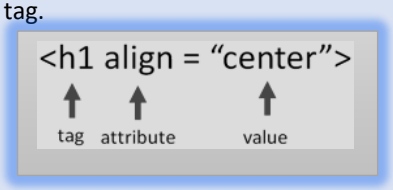
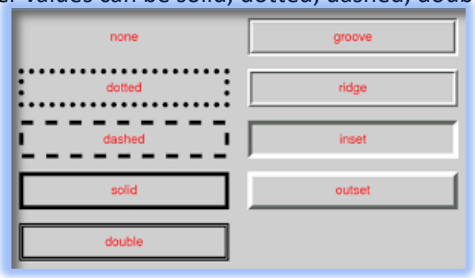
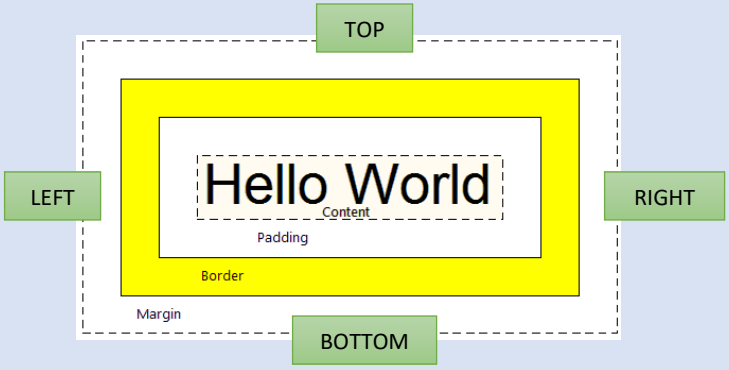
Term	Definition
<!DOCTYPE html>	Document type declaration telling the web browser that this document type is HTML.
<a href>	The <a href> tag is used to define a hyperlink within a webpage. The 'a' stands for anchor and the href is the hypertext reference or URL you are linking to.  *See Hyperlink for more information*.
<blockquote>	Block quotes are used for quoting content from another source. The web browser usually indents block quotes.
** **	Break tags are used to insert a single line break (enter). The tag is an empty tag, meaning it has no end tag.
** **	The , or division tag, is used to organize information on a webpage by group **large sections** of HTML elements together. CSS formatting can then be applied to the entire division. Note: Small groups of text, such as words within a paragraph should be styled with the inline tag. *Also see .*
<head>	The <head> tag holds general information (metadata) about the document such as the title and links to styling sheets. *Not to be confused with <header> or headings. Refer to <header> and headings for more information.*
<header>	The header tag contains introductory content or set of navigational aids for an area or areas on the webpage and can contain: • One or more heading elements (<h1> - <h6>) • Logo or icon • A search form • Authorship information *Not to be confused with <head> or headings. Refer to <head> and headings for more information.*
<hr>	Horizontal rules are horizontal lines that are used to separate content.

Term	Definition
<img>	Image tags are used to add images to the page. Image tags require the first two attributes shown below: • src attribute to specify what image to display. • alt attribute to provide a descriptive text alternative for an image if for some reason a user can not view the image (because of a slow connection, an error in the src attribute, or if the user uses a screen reader. • Title attribute can also be added so that a description will appear when your mouse is over the image. `<img src="image.jpg" alt="flower" title="Orchid Flower">`
** **	Defines a list item
** **	Ordered list (numbered list)
<p>	Paragraph element
<script>	The <script> element is used to embed executable scripts or to reference the link to executable scripts. • JavaScript must be written with <script> tags • <script> tags usually go in the <head> of the HTML document, but may also be placed in the <body>

****	A is used to group one word or a few words together for styling.
	My cat has blue eyes but my turtle has dark green eyes.
	To make the word "blue" blue and the word "dark green" green, as shown above, an inline CSS span should be used:
	`<!DOCTYPE html>`
	`<html>`
	`<body>`
	`<p>My cat has <span style="color:blue; font-weight:bold">blue</span> eyes but my turtle has <span style="color:darkolivegreen; font-weight:bold"> green dark </span> eyes.</p>`
	`</body>`
	`</html>`
	Note: Large sections of text should be styled with the inline tag.
	Also see .

<style>	Style tags are used to define how an element will be displayed. • <style> elements must be placed within the <head> of the HTML document.
** **	Unordered list (bulleted list)

Appendix B: HTML & CSS Vocabulary

Term	Definition
Algorithm	A sequence of steps used to complete a task, usually for a computer to carry out.
Angle Brackets	< > < is the left angle bracket > is the right angle bracket
Attributes	Attributes provide additional information about an element, such as alignment, color, height, and width. • Attributes are always placed in the opening tag. • Attributes are written as: **name="value"**. 
Borders	Borders can be applied to most HTML elements in the body. Border values can be solid, dotted, dashed, double, groove, ridge, inset, and outset. Border color and width can be changed.  
Box Model	The CSS box model is used for element design and layout. Every element is considered a rectangular box with padding, borders, and margins. 

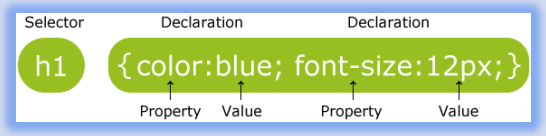
Term	Definition
Class	Class is used when you want make web page styling easier by assigning a class/name to multiple elements throughout the website. You can then apply styles to all of those elements at once.
Code	Programming instructions
Comments	Comments are added to a web page's source code to provide helpful information to anyone that views the source code. • Comments start with an exclamation mark: <! > • Comments do not display in your web pages, they are used to help you and others who design the website. `<!Go back and change the border style for the next two images.>`
Computational Thinking	Using concepts of computer science to solve problems. 1. Computational concepts: sequence, loops, events, conditionals, operators, parallelism, and data. 2. Computational practices: experimenting, iterating, testing, and debugging. 3. Computational perspectives: creating, connecting, and questioning.

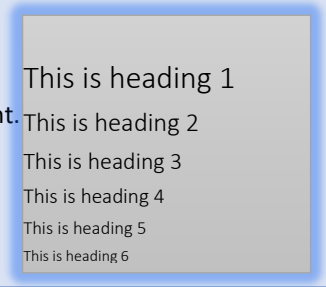
CSS
1. External Style Sheet
	Cascading Style Sheets is a style sheet language used to format the style of a website. CSS controls how the HTML content and web page will look.
	CSS can be written in three different ways:
	With an external style sheet, you can change the look of an entire website by changing just one file. Each HTML page must include a reference (link) to the CSS external style sheet using the <link> element. The <link> element goes inside the <head> section:
	Example <head> element of an HTML document linking to CSS document:
	`<head>`
	`<link rel="stylesheet" type="text/css" href="mystyle.css">`
	`</head>`
	This is an example of the document "mystyle.css" containing the CSS:
	`body { background-color: lightblue; }`
	`h1 { color: navy; margin-left: 20px; }`

2. Inline Style Sheet	For a single element, inline styling can be used but it should be used sparingly as mixing content and styling makes changes more timely. • Inline CSS goes inside the opening tag of the element it is styling: `<h1 style="color:blue;margin-left:30px;"> This is a heading</h1>`

Term	Definition

3. Internal Styles	Internal styles may be used if one single page has a unique style.
	• Internal styles are defined within the <style> element, inside the <head> section of an HTML page:
	
	`<head>`
	`<style>`
	`body {`
	`  background-color: linen;`
	`}`
	
	`h1 {`
	`  color: maroon;`
	`  margin-left: 40px;`
	`}`
	`</style>`
	`</head>`

CSS Syntax	The rules that must be used when coding CSS. In CSS, the syntax rules tell us that we need at least one selector followed by a pair of curly brackets with the properties you are styling for that element, as shown in the example below:  • The **selector** points to the HTML element you want to style. • The **declaration** block contains one or more declarations separated by semicolons. • Each declaration includes a **CSS property name** and a **value**, separated by a colon. • A CSS declaration always ends with a semicolon, and curly braces surround declaration blocks. • No typos or misspellings. • You must have both curly braces. • Declarations must be separated by a semicolon (if you put a colon, it won't work).
Curly Braces *Also known as:* *Curly Brackets*	{ } Curly braces or brackets act as containers for grouping content in CSS. • For every opening curly brace { • there must be a closing curly brace }
Debug	Debugging is the process of locating programming errors and fixing them.

Term	Definition
Element	An element is an individual part of an HTML document. Elements begin with an opening tag, end with a closing tag, and include the content between the two tags.  This is an example of a paragraph element.
Empty Tag *Also known as:* *Empty Elements or* *Self-enclosing* *Elements* *or Void Elements*	Empty tags are tags that don't require content or a closing tag, such as: • for line break • <hr> for a horizontal rule/line *Note: Empty elements can be "closed" in the opening tag like this: . However, HTML5 does not require empty elements to be closed.*
Em Sizes	Font sizes can be defined by "**em**" units. **"em"** is a unit of measure equal to the current font-size. For instance, if the font-size of the document is 12pt, 1em is equal to 12pt; 2em is equal to 24 pt. Ems are becoming more popular in web design due to scalability and their mobile-device friendly nature.
End Tag/Closing Tag	Closing or end tags mark the end of an element. End tags are the same as the opening tag, except that end tags begin with a slash "/" symbol before the element name. Example: </body> to close the body element
Gradient Color	A range of colors that are blended from one color to the next.
Headings **<h1>** **<h2>** **<h3>** **<h4>** **<h5>** **<h6>**	Headings are used to show the document structure so that viewers can quickly skim your pages by its headings. Search engines use the headings to index the structure and content of your web pages so they are important elements. • <h1> headings should be used for **main headings** and is the largest in font size. • <h2>, <h3>, <h4>, <h5>, and <h6> **subheadings** that are progressively less and less important. *Note: DO NOT USE headings to make text BIG or bold.* 
Hexadecimal Code	Hexadecimal codes are used on web pages to define colors. A hex (6 characters) number is written from 0-9 and then A-F. • The first two characters represent the red hue. • The second two characters represent the green hue. • The third two characters represent the blue hue. Examples: White is written as #ffffff while Black is written as #000000.

Term	Definition
HTML	HyperText Markup Language is the computer language used to define the **content** of a web page. Every web page must contain at least some HTML.
HTML Syntax	The rules that must be used when writing in HTML, such as: • No typos or misspellings • Both angle brackets must be used • Opening tags must have their corresponding closing tag • Closing tags must begin with a slash '/' • Elements within elements must be properly nested so that they do not overlap each other. •
HTML Tags	Tags surround content and apply meaning to it, telling web browsers how text, images, and other information should be arranged on a web page. Tags are enclosed in angle brackets < >.
Hyperlink	An icon, graphic, or text in a webpage that links to another file or object when clicked. Links are defined with the <a> tag.  *See <a href> for more information.*
Interactive JavaScript button	JavaScript makes an HTML page more dynamic and interactive. One of the many things you can use JavaScript for is creating buttons that will activate an event when clicked. Interactive JavaScript button types are: clickable button, submit, and reset button.
IT	Information technology--create, manage and exchange information using all types of technology.
Iterative Design Process	The iterative design process is a method used to improve computer programs by constantly cycling through the following steps: 1. Prototype—make the program 2. Test—run the program and test it with different users, noting any errors 3. Analyze—identify the programming changes needed or errors made in the program design 4. Refine—fix any errors in the program
Java	Java is NOT the same language as JavaScript. Java is an object-oriented programming language that is used to create applications that run in a virtual machine or browser. JavaScript is a scripting language whose code is run on a browser only.

Term	Definition
JavaScript	JavaScript is not the same language as Java. JavaScript is a scripting language used to change the behavior of web pages, making them more dynamic and interactive. JavaScript is run on a web browser. JavaScript Syntax Rules to know: • JavaScript is case sensitive. If you accidentally miss a capitalization or capitalize a letter that should not be, the script will not run. • JavaScript must be written within <script> tags. • <script> tags must go in the <head> or <body> of an HTML page. The <head> element is the preferred placement area.

Margins	Margins are used for spacing out elements by setting the size of the **white space outside the border of an element**. There are CSS properties for setting each of the margins: top, right, bottom, and left.
	
	This is an example of CSS for setting the margins around a paragraph.
	
	Margins can be stated as em, %, or px.
	Em's and % will adjust based upon screen size or visual needs.
	Px does not adjust in size, however, it is more predictable.
	
	`p {`
	`  margin-top: 100px;`
	`  margin-right: 100px;`
	`  margin-bottom: 150px;`
	`  margin-left: 80px;`
	`}`

Markup	Means that the code written with standard words that humans can put meaning to. For example, the <body> tag is used to define the body of a web page.
Markup Language	A computer language that uses tags to define elements of a webpage. HTML is a markup language.
Notepad++	Notepad++ is a free software program used as a text editor and source code editor for Microsoft Windows.
Overline	Also known as an overscore or overbar, an overline places a horizontal line directly above the text it is applied to. It is the opposite of an underline.

Padding	Padding is used for spacing out elements by setting the size of the **white space inside the border of an element.** There are CSS properties for setting the padding on each of the sides of an element: top, right, bottom, and left.
	
	This is an example of CSS for setting the padding of a paragraph.
	
	Padding can be stated as em, %, or px.
	Em's and % will adjust based upon screen size or visual needs.
	Px does not adjust in size, however, it is more predictable.
	
	`p {`
	`  padding-top: 50px;`
	`  padding-right: 30px;`
	`  padding-bottom: 50px;`
	`  padding-left: 80px;`
	`}`

Percentage Sizes	Font sizes can be defined by a percentage '$'. The current font size is equal to 100%. Fonts defined as percentages are fully scalable for visually impaired viewers and mobile devices.

Term	Definition
Pixel Sizes	Font sizes can be defined by pixels or px. Pixels are fixed-sized units. Each px is equal to one dot on the computer screen. Although using px for web design is a way to ensure true pixel-perfect representation of a site, a major problem with px is that it does not scale upward for the visually impaired viewers or downward to adjust for mobile devices.
Point Sizes	Font sizes can be defined by points (pt). Each pt is equal to 1/72nd of an inch. Points are fixed-sized units and cannot be scale in size for visually impaired viewers or mobile devices.
Script	A script is code written in a scripting language, meaning that the code will be carried out by another software application. JavaScript is a scripting language because another web browser interprets the code that was written in the text editor, such as Notepad++.
Source Code	Source code is the programming instructions and statements used to create a webpage or computer program. Source code can be viewed and read by humans.
Start Tag/Opening Tag	This tag starts an element. It tells the computer that everything after it is contained within the rules of that tag.
Syntax	Syntax refers to the correct use of a language. Syntax errors occur when the programmer fails to follow the strict rules of the computer language, such as an error in capitalization or spelling. Syntax errors do not occur in Alice because all procedures are written by dragging tiles of code together.
Tags	The parts of the webpage. Tell the computer how to display content. Most often, tags come in pairs: a start/opening tag & an end/closing tag. The standard is to type tags in lower case.
Value	When styling elements, the value of a property can be changed, such as font style, font size, font color, margin width, margin height, and so on. 
Web Page	One page of a website. *Note: web page is 2 words.*
Website	A collection of web pages that are connected together. *Note: website is one word*

Appendix C: Basic Foundation of an HTML Web Page

<!DOCTYPE html> document declaration specifying that this is an HTML5 document
<html> starts the HTML document
<head> contains information about the document (meta information)
<title> specifies a title for the document, which will appear on the web browser's tab
</title> closes the title tag
</head> closes the head tag
<body> contains the main content of the web page
<h1> defines a heading
<p> element defines a paragraph
</body> closes the body of the web page
</html> closes the HTML document. Always the last line of a webpage.

Appendix D: HTML/CSS Units of Measure

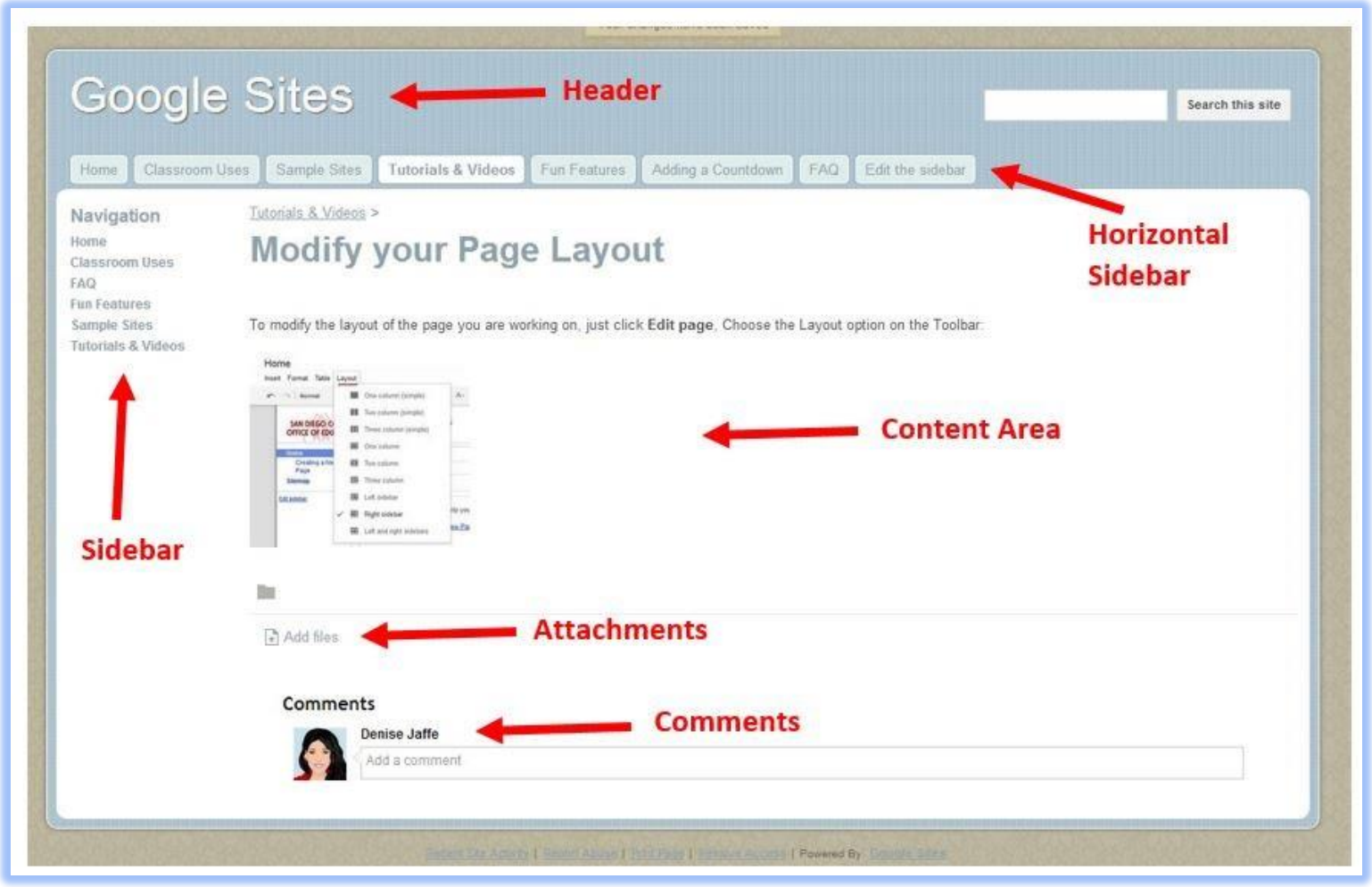
Why are em and % preferred units of measure?

- Point (pt) and pixel (px) values do not rescale in size when the base font-size changes.
- Em (em) and percent (%) units will rescale in size as the base font size changes.

Font size measures comparison

	body { font-size: 100%; }	body { font-size: 120%; }
font-size: 1em	The quick brown fox	The quick brown
font-size: 12pt	The quick brown fox	The quick brown fox
font-size: 16px	The quick brown fox	The quick brown fox
font-size: 100%	The quick brown fox	The quick brown

Appendix E: Google Sites Page Layout

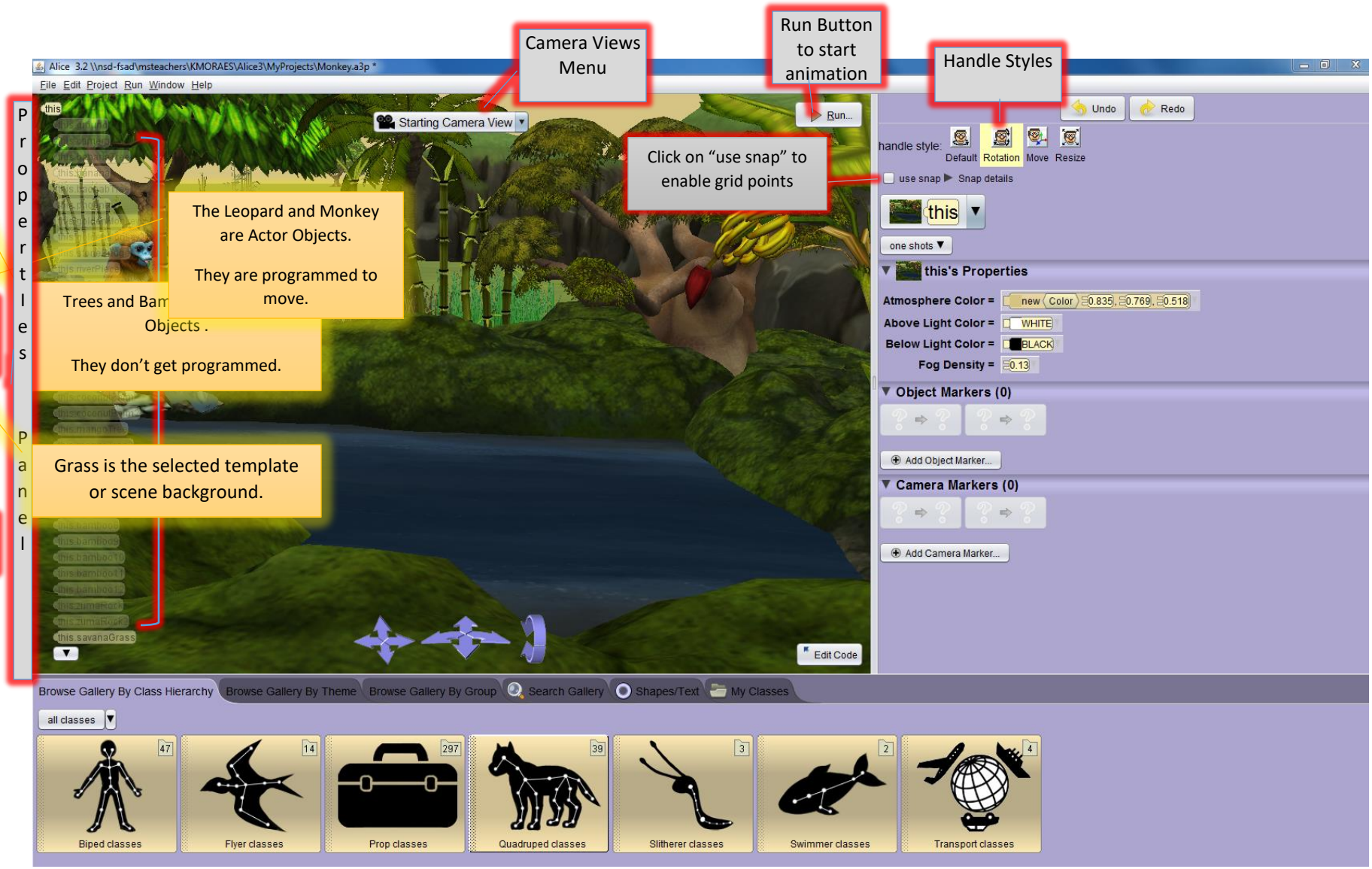


Appendix F: Google Sites Vocabulary

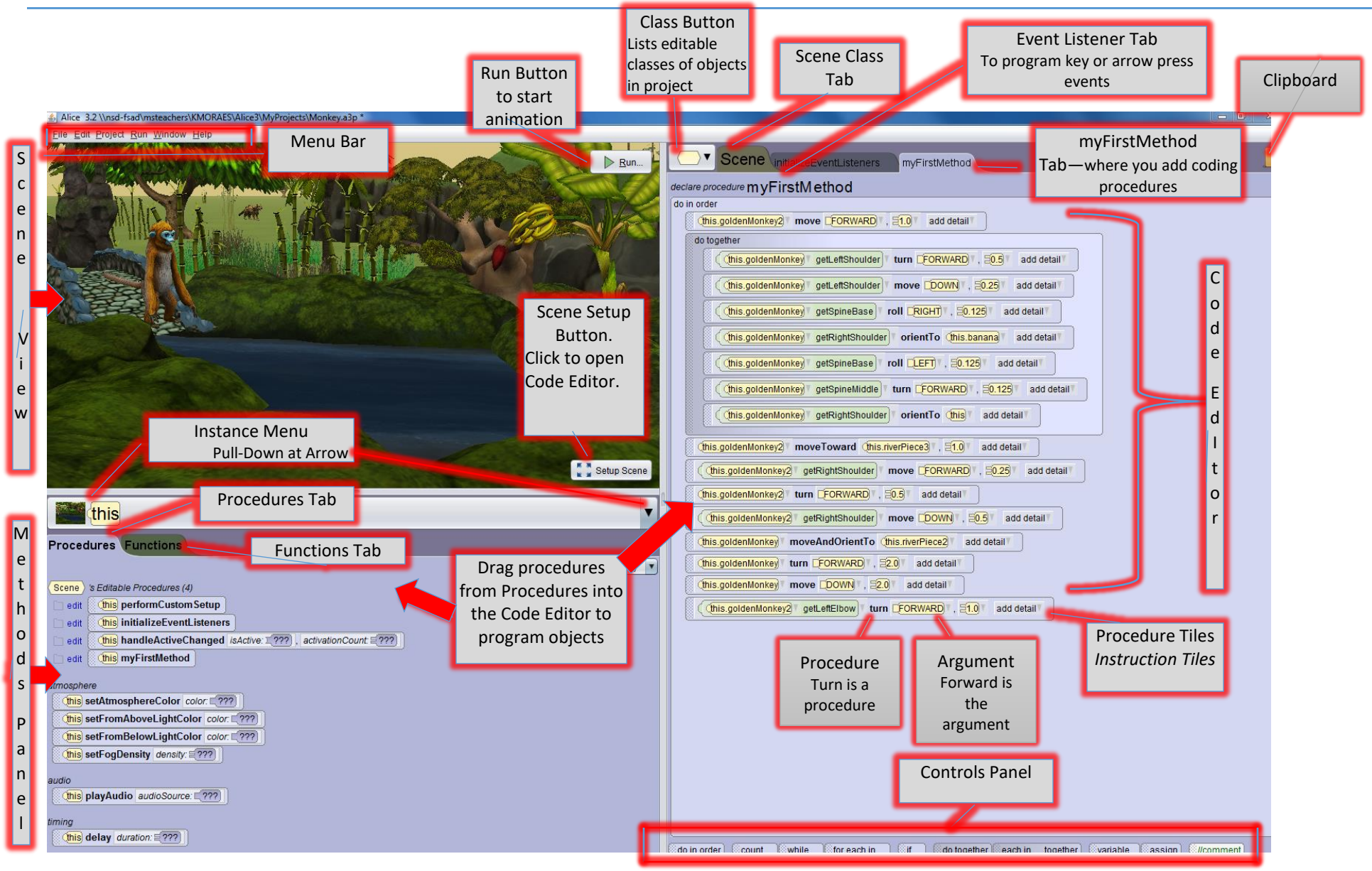
Term	Definition
Attachments	Web page visitors can add files to your web page under the Attachments area. For class, you must turn attachments off.
Comments	Comments allow web page visitors to add comments to your pages. For class, you must turn comments off.
Content Area	The area on the web page that you can change and add content to.
Domain Name	A unique name that identifies a website.
Footer	Information displayed at the bottom of each page.
Gadget	An application that can be added anywhere on the web page. Gadgets appear as small boxes containing some type of useful information that changes with time. There are all types of gadgets, such as # of web page visitors, date and time, current U.S. debt calculator, etc. *Note: Third-party gadgets are developed by someone outside of Google for use on Google Sites. They are not always safe or reliable.*
Header	Information displayed at the top of each page.
Hierarchy	Order of items placed according to ranks.
Horizontal Navigation Bar	Navigation tool displayed across the width of the web page.
Live Collaboration	Google Sites allows multiple users to work on the same website at the same time. Changes can then be merged.
Navigation Panel	Consists of a list of links to the pages in the website.
Page	Each main topic that you want a separate web page for should have its own page.
Private Site	Website with limited access to specific people.
Public Site	Website that is open to anyone with access to the Internet.
Sidebar	Navigation tool displayed across the length of every web page.
Subpage	Subpages are pages that are linked to a main page and fall under the same topic. For instance, you may have a main page "Cupcakes" with subpages "Gluten-free Cupcakes," "No-bake Cupcakes," and "Frozen Cupcakes."

Term	Definition
Template	A web page template is a basic page that can be customized with various types of content.
Theme	A pre-designed style for a website. The font style, font color, background color, and overall color scheme of the site is already set for each area of the web page.
URL	Uniform Resource Locator. The address of a specific website or file.
Web Page	One page of a website.
Website	A collection of web pages that are connected together.

Appendix G: Alice Scene Editor

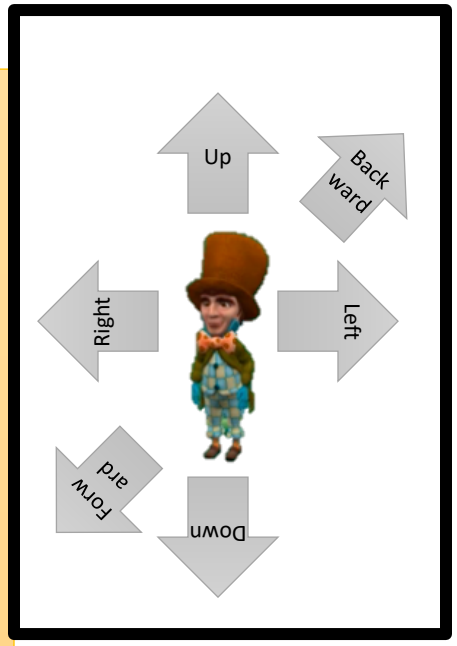


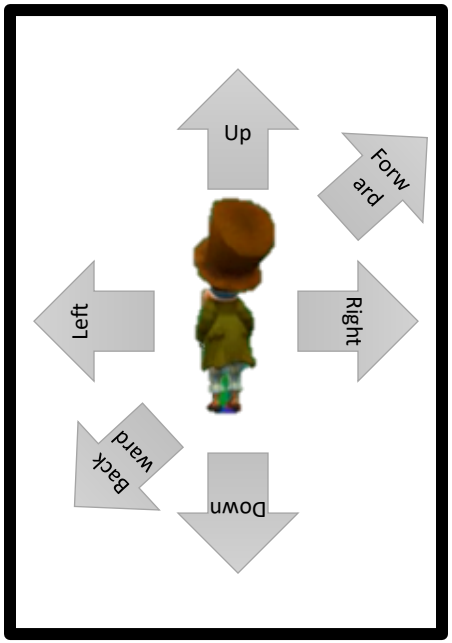
Appendix H: Alice Code Editor

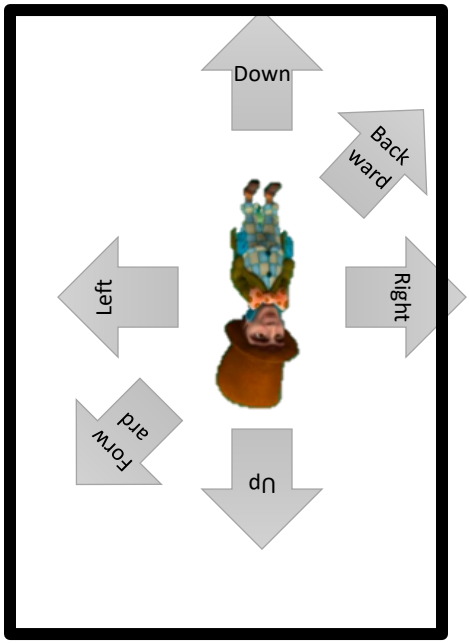


Appendix I: Alice Object Movements

Each object (and its subparts) can move in 6 directions from the object's viewpoint:







Note:

- *All directions are from the object's point of view. For instance, if you want a monkey to move left, it will move to his left--not your left.*
- *Movement values are listed in meters.*
  - *"Move Forward 1" = one-meter movement forward*
- *Turns are listed as revolutions.*
- *Objects rotate around the pivot point (center point), which is where the 3 axes (x,y,z) meet.*
  - *"Turn 1" = 1 full rotation of object or 360 degrees*
  - *".125" = 45 degrees rotation*
  - *".25" = 90 degrees rotation*
  - *".50" =180 degrees rotation*

Center point or pivot point is the position where the object's 3 coordinate axes cross (often called the origin). The center of each object is different but is usually based on the center of the mass.

X-axis: left to right

Y-axis: top to bottom

Z-axis: front to back

Default Handle Style

Use for simple rotation and movement of objects. Click the object to display a ring. Drag the ring to change the direction of the object.

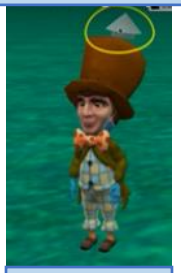
Rotation Handle Style

Use to rotate the object in all directions (x, y, and z coordinates) by clicking on one of the rings that appear.

Move Handle Style

Use to move the object with the arrows that appear (x, y, and z coordinates).

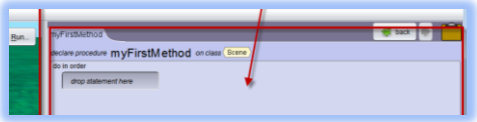
Resize Handle

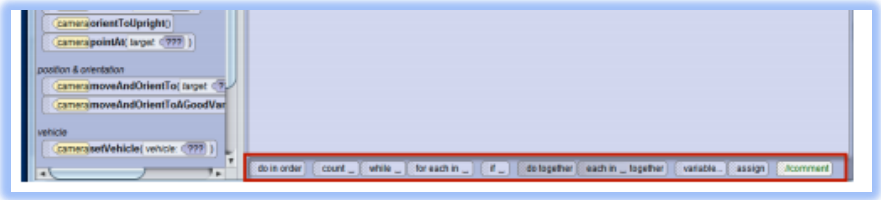


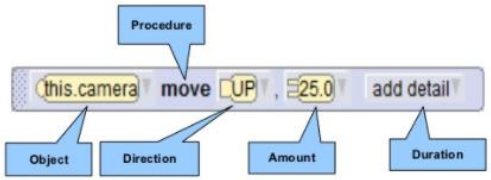
Pull the arrow up or down to resize the object.

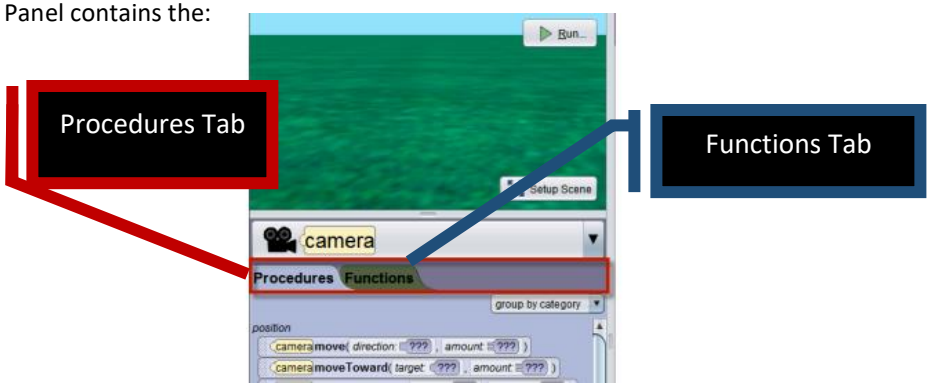
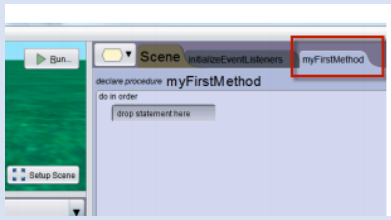
Appendix J: Alice Vocabulary

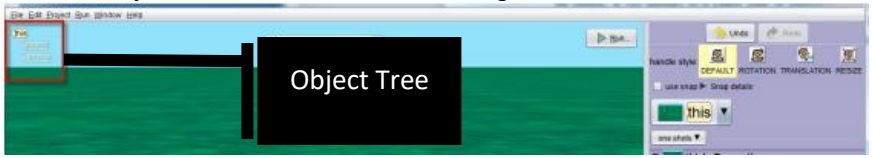
Term	Definition
3D Objects	3D objects have width, height, and depth.
Alice	An innovative 3D programming environment that makes it easy to create an animation for telling a story, creating interactive games, or creating animated videos to share on the web.
Animation	A series of independent movements that, when viewed rapidly, one-after-another, create the illusion of movement.
Argument	An argument is an item of information that must be supplied so that Alice can understand HOW to execute the action. For instance, if you are programming a walrus to turn left, left would be the argument. Without it, Alice would not know where you wanted the walrus to face. Arguments answer questions related to: • Direction • Distance (stated in meters) • Duration (timing) • Text • Target
Axes	To use Alice, you should understand the 3 axes: • X axis: left/right • Y axis: up/down • Z axis: front/back
Background Template	The background where you place the setting and the actor objects.
Billboard	Billboards create a flat box with 2D images on it, such as text.
Boolean Logic	Boolean logic defines conditions, such as “and” “or” or “not” that are necessary for a statement to be executed. If the condition is not met, the statement will not execute an action.
Camera	Alice has only **one camera** in a scene. The camera is used to set the viewing positions in your Alice world.
Camera Marker	A camera marker is an invisible placeholder/marker that saves the camera’s position and orientation. You can create multiple camera views to change the viewing position and zoom in or out of a scene. *Note: Always mark the original position of the camera before creating a new camera marker so that you can go back to that marker when necessary.*

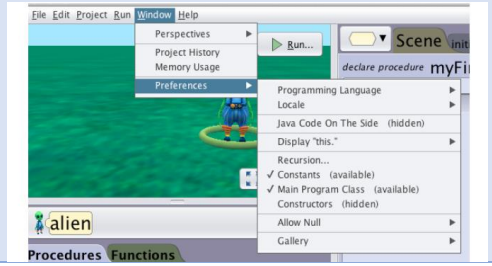
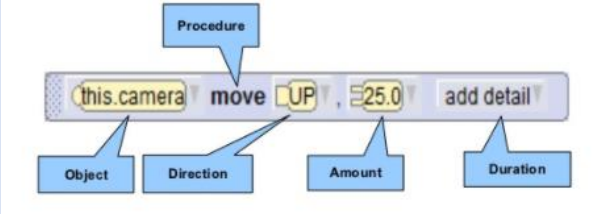
Term	Definition
Camera Move Controls	Move Controls are located in the Scene Editor (at the bottom of the Scene View) and are used to manually position the camera. • Move—the first set of arrows. Moves the camera left, right, up, or down. • Pan—the second set of arrows. Pans (turns) the camera left or right and zooms in/out (moves the camera forward or backward). • Tilt—the third set of arrows. Moves the camera up and down.  • Click and hold an arrow to move or turn the camera in the position of the arrow. • Click and drag in the direction of the arrow to speed up a movement. • Click and drag BETWEEN two arrows and the camera will move in both directions at the same time.
Camera Views Menu	Located in the Scene Editor, the Camera Views Menu provides different perspectives of the scene so that you can view the scene from different viewpoints. This is helpful because it may look like two objects are right next to each other, but in reality, they may be very far apart. Viewing from different angles is needed. • Layout Scene View—provides a high level view of the scene from an angle. • Top View—provides a straight down bird's eye view of the scene. • Side View—provides a view from the side of the original starting camera view. • Front View—provides a straight-on view from the original starting camera view.
Center Point	Each object has its own center point or point that an object rotates around. The center point is usually the center of the mass.
Class (of objects)	A class defines set of similar objects. For example, there are 4 Golden Monkeys (black, brown, gold, and snow) which all belong to the Golden Monkey class. The Golden Monkeys belong to the Biped Class.
Class Hierarchy Menu	Located in the Code Editor, the Class Hierarchy Menu is a drop-down menu displaying all classes in your program.
Clipboard Icon	The clipboard is used for copying procedure tiles into Alice methods. Just drag a block of code up to the clipboard icon in the top right of Alice and then drop it. After navigating to where you want to paste that code, drag the clipboard and then let go where you want to add it.
Code	One or more instructions written in a language that computers understand.
Code Editor Panel	The Code Editor is where you to program your animation. Click on the Edit Code button to display the code editor (it is a toggle button to switch from one editor to the other). 

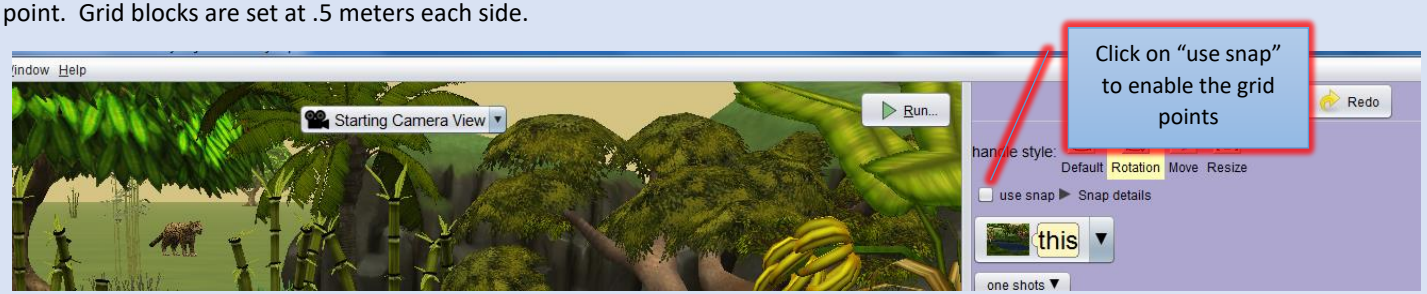
Term	Definition
Command	An instruction for a computer to carry out. To build a program, you create a series of commands such as: Move Up, Say “hello”, Turn, etc.
Computer Science	The study of the ideas, ways of thinking, languages, software and hardware needed to solve problems with computers.
Conditional Statement	A feature of a programming language that performs different computation or actions depending on whether a programmer-specified Boolean condition is either true or false.
Control Statements *Control Tiles*	Located at the bottom of the myFirstMethod Tab, control statements tell Alice **how** to sequence the programming instructions. Control tiles group procedures together so that they run in the selected sequence. • Do in order • Count (a repeat loop) • While • If • Do together • If/else • Comment (used to add programming notations to the coding for future reference). 
Debugging	Finding and eliminating programming errors. Running the animation, making corrections or additions to the programming instructions, and running the animation again.
Editors	In Alice, there are 2 different workspace editors that you will need to toggle between to build your project: 1. Code Editor—where you add programming commands 2. Scene Editor—to set up the scene
Event	Anything a user or "outside force" does to a program--user input. Examples: pressing a specific key or clicking the mouse on an object.
Event Handler	A method describing the actions that objects will do in response to an event.
Execute	To carry out or **run** a command or set of instructions.
Files Menu	The Files Menu is used for managing Alice files. 
Function	A method that asks a question, returning a value, such as the distance between two objects.

Term	Definition
Functions Tab	In the Scene Editor, the Functions Tab is the area where all pre-defined functions for objects are listed.
Gallery	At the bottom of the Scene Editor, the Gallery is a collection of objects that can be inserted into the scene. You can browse by: Class Hierarchy, Theme, Group, Search, and Shapes/Text.
Handle Styles	In the Scene Editor, there are 4 Handle styles used to move objects: 
Imprecise Positioning	Positioning objects by dragging them into the Scene View and dropping them where you want them to appear.
Initial Scene	The first scene of an animation, containing three main elements: 1. A background template, which provides the sky, ground, and light. 2. One or more non-moving scenery objects, which provide the setting. 3. One or more moving objects, which provide the action.
Instance	Each individual, specific object. If you have 3 chickens in the scene, each one is its own instance and can be programmed separately from the other chickens.
Instance Menu	Located in the Scene Editor, the Instance Menu is a pull-down menu listing all of the objects to program.
Instruction	A method name and its arguments (assigned values).
Instruction Tiles ***Procedure Tiles***	The blocks of code that are dragged and dropped to make a program are instruction tiles. 
Iteration	The repetition of a sequence of computer instructions for a sepcified number of times or until a conition is met.
Iterative Design Process	A process used to program in which you continually develop, test, and fix the programming errors: 1. Prototype 2. Test 3. Analyze 4. Refine 5. Repeat
Keyboard Listeners	Keyboard controls allowing the user to control one or more objects while the animation is running. For example, the arrow keys.

Term	Definition
Logic Error	Most errors in Alice are logic errors. Logic errors are the most difficult to find and to fix because there is no obvious error. A logic error would be one in which the program runs but does not do what you wanted it to do. Logic errors occur because the programmer does not understand the end result of a procedure. For instance, if you program a dolphin to turn 85 degrees but instead it turns 85 complete revolutions, the logic error is that turns are stated in revolutions, not degrees so you would have to convert 85 degrees into revolutions (85/360). To prevent logic errors: • Use the iterative design process so that you can find and fix errors before you add all of the coding • Think before coding
Loop	A programming instruction that repeats a sequence of instructions. In Alice, the repeat loop is a Control Statement/Tile, "Count."
Menu Bar	The menu bar is located at the top left corner of the window. 
Method	A series of instructions given to objects to describe actions to take, such as "say," "move," "set opacity," etc.
Methods Panel	The Methods Panel contains the: 
myFirstMethod	 Located in the Scene Editor, myFirstMethod is the area where programming statements are placed so that they will execute when the "RUN" button is clicked. myFirstMethod is automatically selected as the active editor. "Drop statement here" is where you drag your programming instruction for each instance (object).
Nesting	Placing one programming statement inside of another.

Term	Definition
Object	Any sort of three-dimensional shape such as a person, animal, building, etc. in the Alice World. Three objects are in EVERY project by default: 1. Ground 2. Camera 3. Light
Object Markers	An invisible placeholder/marker used to record the location and orientation of objects in the scene.
Object Menu	In the Code Editor, you can view the Object Menu to see a list of all the objects in a scene. 
Object Tree	Shows the objects in the current Alice world organized as a list of tiles. 
Object-oriented programming	OOP is the most modern type of computer programming based on objects rather than actions and data rather than logic. Alice and Scratch are both object-oriented programming languages.
One-shot Procedures	Used to make scene adjustments and position objects prior to any animated movements. One-shot procedures DO NOT run when the program is executed. Use one-shot procedures to place the objects in the initial scene.
Opacity	Pronounced: opacity: oh-pas-i-tee (a soft A). How easy it is to see through an object. A value of zero makes the object invisible; a value of 1 makes the object fully visible.
Orientation	The relative position and direciton of an object. Orientation is important because it will determine which direction is forward/backward, up/down, and left/right. In Alice all movements are made in relation to an object's own orientation (left is to their left, not your left).
Pair Programming	Two developers or students team together and work on one computer. Roles: "Driver" "Navigator"/Switch roles each day.
Precise Positioning	Positioning an object in by setting new values for the x, y, and z coordinates. The object will automatically reposition to the coordinates entered. *Note: You must hit the enter key after entering the values to have the object move to that location.*

Term	Definition
Preferences Menu	On the Menu Bar, click on Window and then go down to Preferences to access the Preferences Menu.  If you want to see the Java coding, select "Java Code on the Side"
Procedural Abstraction	The concept of making code easier to understand and reuse. For instance, programmers in Alice can declare (create) a new procedure that can be used in multiple statements and even in other objects.
Procedures	Methods that perform an action. In Alice, procedures are programmed by dragging procedure tiles into the myFirstMethod area.
Procedures Tab	Located in the Methods Panel, displays procedures or actions that the object can be programmed to do. Programming instruction tiles are dragged into the myFirstMethod tab of the Code Editor in order to program objects.
Procedures Tiles ***Instruction Tiles***	Blocks of coding instructions or actions for objects to perform. 
Properties Tab	In the Scene Editor, the Properties Tab is where all properties of an object are listed. An object's properties can be altered here.
Roll	Tilting the object's center point left or right.
Run Button	Launches the runtime window and starts the animation.
Run Time Errors	Run time errors occur whenever the program instructs the computer to do something that it is not capable of doing or is unwilling to do. • Statements may be in the wrong order • Procedures may not make sense, such as programming an object to move forward 0 (zero) meters.
Scene Editor	The Scene Editor is the screen in Alice used to select objects for your scene and set up opening positions. 

Term	Definition
Setting	Objects that will remain stationary in your animation. Examples: a tree, rocks, or a building.
Six Directions	In Alice, objects can move in six different directions: up, down, left, right, forward, and backward.
Snap Grid	In the Scene Editor, you can check the box next to "use snap." Dragging the object will cause the object to snap into position at the nearest grid point. Grid blocks are set at .5 meters each side. 
Statement	A written instruction that can be one of 3 types: • Action to be performed • Control to determine the sequence in which actions are performed • Comment statements that the programmer inserts as programming notes
Sub-Parts of an Object ***Joints***	Most objects have subparts such as (head, neck, right hip, left hip). These subparts are connected with joints. By rotating the joins, you are able to reposition the subparts. Internal joints that can be moved into different positions. To view the joints of an object: • In the Code Editor, select the object from the Object Menu and drag over to the subpart menu arrow, selecting the joint. The procedures for that joint will then be displayed in the Detail Panel and can be used to create an animation. • Select the object from the Objects Menu. Click the Show Joints checkbox. Each subpart has its own orientation and they move, turn, and roll based upon that orientation. • The joints of an object can also be selected in the Scene Editor in the Object Menu. Click the down arrow to open the Object you want to change and then pull the mouse over to the right arrow to open a cascading menu of joints. When a joint is selected, Alice automatically displays the rotation handles around the selected joint. The rings can be used to rotate the subparts into the desired position. • One-shots can also be used to rotate the subparts.
Syntax	Rules that must be used for a computer language. Each language has its own syntax/rules. The grammar, structure, or order of the elements in a programming language are all part of the syntax. If you do not use the proper syntax, the computer will not know what to do.
Tiles	Blocks of coding instructions in Alice.
Translation	The Handle Style, Translation, has been replaced with the Move Handle. *See Handle Styles for more information.*

Term	Definition
Troubleshooting	A systematic approach to problem solving that is often used to find and resolve a problem, error, or fault within a program.
Undo & Redo Buttons	Each click of the Undo button reverses the most recent action by removing the changes made. It is possible to Click Undo repeatedly, backtracking all the way to the beginning of a project or until your last save. If you want to reverse a change you made, use the UNDO button to reverse one change at a time. If you hit UNDO too many times and want to go back, use the REDO button.
User-defined procedures	Users can create, “declare,” new procedures so that the procedure can be used multiple times without having to add every procedural event. For instance, you can create an “elephantwalking” procedure and use it anytime you want the elephant to walk within the project.
User-generated events	Events that are triggered by keystrokes or mouse clicks. An event listener must be programmed for user-generated events.
Vehicle	All Alice objects are attached to a "vehicle." All movement by the object is done in relation to that vehicle. If a vehicle moves, the object moves with it. You can change an object’s vehicle. For instance, if you want to put a person inside of a moving racecar, you would click on the person, go to the Properties Tab and set the vehicle to the racecar. Then, when the car moves, the person moves with it.
World	In Alice, a World encompasses everything that exists in the project. Every object, method, event, and even comments lie inside the world in Alice.

Appendix K: Alice Procedures & Arguments

Detail/Procedure	Value	Description
animationStyle	*BEGIN_AND_END_ABRUPTLY* *BEGIN_GENTLY_AND_END_ABRUPTLY* *BEGIN_ABRUPTLY_AND_END_GENTLY* *BEGIN_AND_END_GENTLY*	BEGIN_AND_END_ABRUPTLY starts at top speed and ends with a sudden stop. BEGIN_GENTLY_AND_END_ABRUPTLY begins with a gradual acceleration to top speed and ends with a sudden stop. BEGIN_ABRUPTLY_AND_END_GENTLY starts at top speed and ends with gradual deceleration. The default animation style is BEGIN_AND_END_GENTLY, which begins with a reasonable period of acceleration, then constant movement at some top speed, followed by a reasonable period of deceleration.
Duration	*DecimalNumber*	By default, Alice animation methods execute in 1 second. Changing the duration will change the number of seconds.
orientTo	*Target Object*	Animates a rotation of the object around its pivot point, so that its orientation will be exactly the same as the orientation of the target object. The object's position will be unchanged.
orientToUpright		Animates a rotation of the object around its pivot pint, so that its sense of up will be perpendicular to the ground.
pointAt	*Target Object*	Animates a rotation of the object around its pivot point, so that its sense of forward will be in the direction of the target's pivot point.
Roll	*Direction, DecimalNumber*	Animates a roll of the object around its pivot point, in the specified direction according to its own orientation, by the specified amount, given in fractional parts of a full rotation. The object's sense of forward will remain unchanged during the animation.
setAmbientLightColor	*Color*	Sets the color of the primary light source in 'this' scene. Think of it as the color of sunlight in an outdoor scene.
setAtmosphereColor	*Color*	Sets the color of the sky in 'this' scene.
setFogDensity	*DecimalNumber*	Used to set the density of the fog in 'this' scene by setting the density value in the range of values from 0.0 (no fog) to 1.0 (full fog with no visibility of objects in the fog).
setFromAboveLightColor	*Color*	Sets the color of a secondary light source from above in 'this' scene.
setFromBelowLightColor	*Color*	Sets the color of a second light source from below 'this' scene.

Detail/Procedure	Value	Description
setOpacity	*OpacityNumber*	Sets the transparency of 'this' object by setting the opacity value of 'this' object using a range of values from 0.0 (invisible) to 1.0 (fully opaque/visible).
setPaint	*Paint*	Sets the paint value of 'this' object to the paint argument.
Turn	*Direction, DecimalNumber*	Animates a turn of an object around its pivot point (center point), in the specified direction according to its own orientation, by the specified amount, given in fractional parts of a full rotation. The object's sense of forward will be changing during the animation.
turnToFace	*Target Object*	Animates a turn of the object around its pivot point, so that its sense of forward will be in the direction of the target object.