Name: Robert Lefrandt	Grading Quarter: 1	Week Beginning: 09/02/2024
School Year: 2024-25	Subject: Software & App Design	

Monday	Notes: Minecaft for Education (Python) Amazon Future Engineers (AFE) (Python) Kahn Academy Microsoft Visual Code for Educators Python	***No School*** Students will: • continue to define what the Software & App Class is and what are the Arizona State Standards, skills, and possible credentials, certifications. • Understand the front and back end of a web-stack • Recognize various programming **Learning Management Systems** (LMS) • Aware of other programming resources Lesson Overview: Online Courses, LMS examples: Learn JavaScript and Python • Start with WebStack: front-end: HTML , CSS. JavaScript using freecodecamp • Create accounts for freecodecamp.org ◦ HTML ◦ Cascading Style Sheets (CSS) ◦ Resources: ▪ w3schools.com ▪ Stackoverflow.com • TechSmart: CS Python Learning Management System (LMS) ◦ Login to online Python Student Accounts ◦ www.techsmart.codes/ ◦ **Lesson 1.1: Statements and Variables** ◦ **Lesson 1.2: Libraries** ◦ **lesson 1.3: (Data Type) Integers & Floats** ◦ resources: pythontutor	Academic Standards: **Arizona CTE: Software & App Design** **11.0202.00** **Technical Standards** **STANDARD 12.0 DEVELOP A PROGRAM** 12.1 Use a program editor to enter and modify code 12.2 Identify correct input/output statements

Tuesday	Notes: Minecaft for Education (Python) Amazon Future Engineers (AFE) (Python) Kahn Academy Microsoft Visual Code for Educators Python	Students will: • continue to define what the Software & App Class is and what are the Arizona State Standards, skills, and possible credentials, certifications. • Understand the front and back end of a web-stack • Recognize various programming **Learning Management Systems** (LMS) • Aware of other programming resources Lesson Overview: Online Courses, LMS examples: Learn JavaScript and Python • Start with WebStack: front-end: HTML , CSS. JavaScript using freecodecamp • Create accounts for freecodecamp.org ◦ HTML ◦ Cascading Style Sheets (CSS) ◦ Resources: ▪ w3schools.com ▪ Stackoverflow.com • TechSmart: CS Python Learning Management System (LMS) ◦ Login to online Python Student Accounts ◦ www.techsmart.codes/ ◦ **Lesson 1.1: Statements and Variables** ◦ **Lesson 1.2: Libraries** ◦ **lesson 1.3: (Data Type) Integers & Floats** ◦ resources: pythontutor	Academic Standards: **Arizona CTE: Software & App Design 11.0202.00 Technical Standards** **STANDARD 12.0 DEVELOP A PROGRAM** 12.1 Use a program editor to enter and modify code 12.2 Identify correct input/output statements

Wednesday	Notes: Minecaft for Education (Python) Amazon Future Engineers (AFE) (Python) Kahn Academy Microsoft Visual Code for Educators Python	Students will: • continue to define what the Software & App Class is and what are the Arizona State Standards, skills, and possible credentials, certifications. • Understand the front and back end of a web-stack • Recognize various programming **Learning Management Systems** (LMS) • Aware of other programming resources Lesson Overview: Online Courses, LMS examples: Learn JavaScript and Python • Start with WebStack: front-end: HTML , CSS. JavaScript using freecodecamp • Create accounts for freecodecamp.org ◦ HTML ◦ Cascading Style Sheets (CSS) ◦ Resources: ▪ w3schools.com ▪ Stackoverflow.com • TechSmart: CS Python Learning Management System (LMS) ◦ Login to online Python Student Accounts ◦ www.techsmart.codes ◦ **Lesson 1.1: Statements and Variables** ◦ **Lesson 1.2: Libraries** ◦ **lesson 1.3: (Data Type) Integers & Floats** ◦ resources: pythontutor	Academic Standards: **Arizona CTE: Software & App Design 11.0202.00 Technical Standards** **STANDARD 12.0 DEVELOP A PROGRAM** 12.1 Use a program editor to enter and modify code 12.2 Identify correct input/output statements

Thursday	Notes: Minecaft for Education (Python) Amazon Future Engineers (AFE) (Python) Kahn Academy Microsoft Visual Code for Educators Python	Students will: • continue to define what the Software & App Class is and what are the Arizona State Standards, skills, and possible credentials, certifications. • Understand the front and back end of a web-stack • Recognize various programming **Learning Management Systems** (LMS) • Aware of other programming resources Lesson Overview: Online Courses, LMS examples: Learn JavaScript and Python • Start with WebStack: front-end: HTML , CSS. JavaScript using freecodecamp • Create accounts for freecodecamp.org ◦ HTML ◦ Cascading Style Sheets (CSS) ◦ Resources: ▪ w3schools.com ▪ Stackoverflow.com • TechSmart: CS Python Learning Management System (LMS) ◦ Login to online Python Student Accounts ◦ www.techsmart.codes/ ◦ **Lesson 1.1: Statements and Variables** ◦ **Lesson 1.2: Libraries** ◦ **lesson 1.3: (Data Type) Integers & Floats** ◦ resources: pythontutor	Academic Standards: **Arizona CTE: Software & App Design 11.0202.00 Technical Standards** **STANDARD 12.0 DEVELOP A PROGRAM** 12.1 Use a program editor to enter and modify code 12.2 Identify correct input/output statements

Friday	Notes: Minecaft for Education (Python) Amazon Future Engineers (AFE) (Python) Kahn Academy Microsoft Visual Code for Educators Python	Students will: • continue to define what the Software & App Class is and what are the Arizona State Standards, skills, and possible credentials, certifications. • Understand the front and back end of a web-stack • Recognize various programming **Learning Management Systems** (LMS) • Aware of other programming resources Lesson Overview: Online Courses, LMS examples: Learn JavaScript and Python • Start with WebStack: front-end: HTML , CSS. JavaScript using freecodecamp • Create accounts for freecodecamp.org ◦ HTML ◦ Cascading Style Sheets (CSS) ◦ Resources: ▪ w3schools.com ▪ Stackoverflow.com • TechSmart: CS Python Learning Management System (LMS) ◦ Login to online Python Student Accounts ◦ www.techsmart.codes/ ◦ **Lesson 1.1: Statements and Variables** ◦ **Lesson 1.2: Libraries** ◦ **lesson 1.3: (Data Type) Integers & Floats** ◦ resources: pythontutor	Academic Standards: **Arizona CTE: Software & App Design 11.0202.00 Technical Standards** **STANDARD 12.0 DEVELOP A PROGRAM** 12.1 Use a program editor to enter and modify code 12.2 Identify correct input/output statements