Lesson Planning Template **Teacher: DeSIRE**

📆**Date:**	**Topic: Introduction to Coding**	**Time Required: 30 minutes**

🎯**Learning Target/Objectives:**

- I can describe the parts and function of a computer
- I can differentiate between a computer and a microcontroller

📜**Vocabulary:**

- Coding
- Programming
- Computer
- Microcontroller
- Arduino
- Hardware
- Software
- Open Source
- Operating System
- App (Application)

Color Legend:
Red = Essential
Blue = Enrichment
Purple = In Depth

❓**Guiding Questions:**

- Differentiate between Coding and Programming
- Identify the physical and tangible parts of a computer and describe their functions
- Differentiate between operating systems and applications

📚**Lesson Design Details:**
Activity 1: Students will identify various computers in the learning environment and how their different designs allow for the execution of the purposes they serve.

Activity 2: Students will examine a microcontroller (Arduino Uno), describe its functions and explain when a microcontroller might be more appropriate for a task than a computer and vice versa.

Key Points:
What is a computer (aka computing device/system)? A machine that can store, process, and manipulate data, or information. Can be programmed to perform a variety of tasks, such as: calculate data (arithmetic, logical operations), execute algorithms, displaying information, produce audio, interacting with humans in a human voice, store, and analyze information.

A computer requires hardware and software to operate.

- **Hardware** - The physical and tangible parts of the computer (e.g., CPU, hard disk, monitor, keyboard, and mouse).
- **Software** - The programs that run on a computer (e.g., operating system, applications). The software is intangible (unable to be touched or grasped). Software tells the hardware what to do and how to do it.

What is code? Computer code is a set of instructions that tells a computer how to perform tasks.

What is software? Computer software is a set of instructions that tells a computing device (i.e., desktop, laptop, tablet, single unit processor) what to do and how to do it.

Who are computer coders and programs? People who write and create software code that tells a computer how to perform tasks. They write and use programming languages to create websites, computer software applications (apps), and other software. They also design software architecture, test systems, manage databases, and debug code.

Computer Coding vs. Computer Programming

- Computer coding and programming are not exactly the same thing. Coding is only one small part of programming.
- Coding is the process of allowing humans to speak to computers.
- Computers can only understand binary language (a series of zeros and ones).
- Humans need to use programming languages to translate and give computers instructions.
- Computer Programming is science and has a wider scope of operations than coding. Programming includes the planning and development of the specific software product application, testing the product, deploying the product, and providing service and maintenance for the product.

- **What is an Arduino microcontroller?** Arduino is an open-source circuit board and computing platform used for building and controlling electronics projects. There are many models of the Arduino brand. The DeSIRE program uses the UNO model. The Arduino consists of both a physical programmable circuit board (also known as a microcontroller) and software (IDE - Integrated Development Environment) that runs on your computer.

Advanced Coding

The DeSIRE Advanced Manufacturing course defines “Advanced Coding” as writing code to perform complex and specific tasks and applications related to advanced manufacturing. Also, it helps students understand basic computer science knowledge which encompasses coding and software programming. The topic goal is to help students become better coders and increase their knowledge about various areas of computer science. Some learning should include:

- Write and use code to control complex circuits or technologies (robotics, sensors)
- Able to read and interpret code functions and operations
- Make a flowchart or handwritten diagram of the code operations or functions
- Rewrite (Repurpose) code for a different purpose to save time or make improvements
- Type many lines of code without spelling or syntax error
- Debug error messages quickly and without help
- Use complex commands to reduce lines of code (e.g., If statements, variables,)

Help and support other classmates with coding questions and concerns

📝**Materials/Resources:**

- Digital Journal (Google Slides RECOMMENDED):

📋**Closing (Check for Understanding):**

- Discussion Review - students will share

- Answers to Guiding Questions
- Any surprises they experienced

Category	Standard Organization	Standard/Benchmark Code & Description
Science	**NCSOS**	**6.P.3.1:** Explain how to measure mass, length, and volume of different materials.
		7.P.2.1: Explain how molecule movement is affected by temperature, pressure, and volume (contextualized through smelting and forging in manufacturing).
		8.P.1.2: Explain the law of conservation of energy regarding energy transformation (e.g., water mills or steam engines).
		8.E.2.2: Explain energy source consequences and environmental impacts (e.g., the shift from wood to coal).
Mathematics	**NCSOS**	**6.RP.A.3:** Use ratio and rate reasoning to solve problems (e.g., calculating historical production rates).
		7.RP.A.2: Recognize and represent proportional relationships (e.g., cost analysis over time).
		8.EE.B.5: Graph proportional relationships and interpret unit rate as slope (e.g., visualizing historical production data).
Technology & Engineering	**ITEEA (STEL)**	**Standard 2:** Core Concepts of Technology and Engineering (understanding systems and resources).
		Standard 4: Impacts of Technology (evaluating positive and negative effects on society).
		Standard 5: Influence of Society on Technological Development (how needs/wants shape tools).
		Standard 6: History of Technology (understanding historical eras defined by tech).

Digital Literacy	**ISTE**	**1.3 Knowledge Constructor:** Students employ research strategies and curate digital resources to build knowledge.
		1.5 Computational Thinker: Students use algorithmic thinking to understand systems (contextualized by deconstructing manufacturing processes).
		1.7 Global Collaborator: Students work effectively in teams to explore local and global issues.
Computer Science	**NCSOS**	**6.AP.1.1:** Develop an algorithm to complete a task (sequential manufacturing instructions).
		7.IC.1.1: Understand the impact of computing/technological innovations on society and economy.
		8.DA.1.1: Collect, organize, and summarize data (e.g., tracking inventory or production output).