**Integrated Development Environment**

1. List five main features of a software integrated development environment (IDE)

Explorer, code editor, Data file viewer, [compiler](http://whatis.techtarget.com/definition/compiler)/[interpreter](http://whatis.techtarget.com/definition/interpreted-script) and a debugger

1. For each main feature listed in #1 above, explain the feature and how the Arduino Create environment provides this feature.

**Code Editor:** The source code editor is a text editor program designed specifically for editing source code of computer programs by programmers. It may be a standalone application or it may be built into an integrated development environment (IDE). Arduino’s program environment allows you to edit and create your own code.

**Explorer**: The Explorer Filesystems tab presents a directories and files that have been set to use in the system. Files that are used by it or one of their accessory programs, such as AppBuilder, are identified by unique icons. Clicking on one of these files with the left mouse button selects it, while clicking with the right mouse button opens a popup menu displaying the operations available for that file. The Arduino Create Environment files of code can be store within the cloud and then by right-clicking allows the use available operations.

**The Data File Viewer:** The Data File Viewer provides a convenient way to view the contents of database files. The Data File Viewer uses the Filesystem Server to open and parse database files, which means that Services must be running in order to use it. Saved code files can be viewed on the left bar opened from the sketchbook tab.

[**Compiler**](http://whatis.techtarget.com/definition/compiler)**/**[**interpreter**](http://whatis.techtarget.com/definition/interpreted-script)**:** An interpreted program, sometimes called a script is a program whose instructions are actually a logically sequenced series of operating system commands, handled one at a time by a command interpreter. In turn, the command interpreter requests services from the operating system. The code check system in Arduino works to verify the code of any errors and if any resulting errors are displayed as highlighted text rather than hyperlinks in many other IDE softwares.

**Debugger:** The Debugger incorporates debugging and editing tools into the environment, adding the ability to work with multiple files and manage projects. The Debugger features color syntax highlighting and code completion like the Source Editor. Just like in the Arduino environment, the Debugger is always connected to an interpreter and performs automatic syntax error checking as code is modified. The Debugger good for verifying application code and making minor editing corrections.

**Version Control System**

1. List five main features of a software version control system.

Git, File sharing, easy access, file recovery and file recovery (history).

1. For each main feature listed in #3 above, explain the feature and how the GitHub environment provides this feature.

**Git:** is a version control system for tracking changes in computer files and coordinating work on those files among multiple people. In Github it can be done with the use of one account as in most cases these files are interacted with by one user.

**Easy Access:** A resource/program that is user friendly and easy to access. Github is both of those things and can be used from anywhere where there is internet.

**File sharing**: The practice of distributing or providing access to digital media, such as computer programs, multimedia (audio, images and video), documents, etc… Github allows the creation repositories that incorporated the use of uploading files for people such as your instructor to see.

**File recovery**: The process of rebuilding or recovering lost files from a disk or hard drive that is no longer operational or has been damaged from unnatural causes. GitHub can be used as a place to backup data/files and if disk/hardware are corrupt. This will be a good place where then can be recovered.

**File recovery (history)**:

The process of rebuilding or recovering lost files from a disk or hard drive that is no longer operational or has been corrupted. GitHub can be used as a place to backup data/files and if disk/hardware are corrupt. As data is always stored in the clouds history and is only removed from your selection.

1. Explain any version control features that we have not made use of in the class so far but that would be useful in the future.

**Pull/update/sync speed**: The speed at which we can pull/update/sync changes out of a remote server repository. The speed at which we can commit/push changes back to that remote server

**Code Review:** A Built-in review tools system to review code.

Others that we could incorporate more into such as History, recovery, find file, and projects.

**Programming Errors**

1. Define and explain a “syntax error” when programming code.

An error in a program due to a code that does not function to the order expected by the programming language. (A typo/typing error)

1. Create a sample Arduino program that has a syntax error. Answer this question by copying and pasting your sample code below and by providing an explanation.

void setup()

// initialize digital pin LED\_BUILTIN as an output.

pinMode(LED\_BUILTIN, OUTPUT);

}

// the loop function runs over and over again forever

void loop() {

digitalWrite(LED\_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)

delay(1000); // wait for a second

digitalWrite(LED\_BUILTIN, LOW); // turn the LED off by making the voltage LOW

delay(1000); // wait for a second

}

“Expected initialize before pinmode.” Indicates that there was suppose to be a semicolon at the start and end of the void set up. In this case the first semicolon was missing.

1. Define and explain a “runtime error” when programming code.

A runtime error is a program error that occurs while the program is running.

1. Create a sample Arduino program that has a runtime error. Answer this question by copying and pasting your sample code below and by providing an explanation.

void setup() {

int status;

pinMode(13, OUTPUT);

digitalWrite(13, status);

In the code that still verifies “status” is placed instead of “OUTPUT” or “greenLED = 13”. When tested on the Arduino board the LEDs don’t turn on.

1. Define and explain a “logic error” when programming code.

Is a mistake in a program's source code that results in incorrect or unexpected behavior.

1. Create a sample Arduino program that has a logic error. Answer this question by copying and pasting your sample code below and by providing an explanation.

**error example:**

int led = 13;

void setup() {

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

}

**fixed example:**

int led = 13;

void setup() {

pinMode(led, OUTPUT);

}

If the LED is turning on very dimly or, you may be missing a pinMode(led, OUTPUT); statement like the error example above. If missing a pinMode statement for a omponent, the code will not work at all, or will barely work. For example, if the component is an LED it will only turn on very dimly. As from the Arduino Environment it can be verified but will have incorrect results.