**Integrated Development Environment**

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

* The Explorer
* The Source Editor
* The Debugger
* The Compiler
* The Data File Viewer

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

* **The Explorer-** The **Explorer** Filesystems tab presents a hierarchical view of directories and files that have been mounted for use in NetBeans.
* **The Source Editor-** The **Source Editor** is a modern programmer's text editor that is optimized for writing Business BASIC source code.
* **The Debugger-** The **Debugger** incorporates BBj debugging and editing tools into the NetBeans environment, adding the ability to work with multiple files and manage projects.
* **The Compiler-** Allows the compiling of Basic Source code into different token formats
* **The Data File Viewer-** The Data File Viewer provides a convenient way to view the contents of database files.

**Version Control System**

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

* Size
* Backup and Restore.
* Synchronization.
* Long-term undo and Short Term Undo
* History

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

* **Size- Unlimited space for repository.**
* **Backup and Restore- Files are saved as they are edited**
* **Sharing- Let’s people share files and stay up-to-date with the latest version.**
* **Short-term undo and Long term undo- If you download a file you handed in and needed to make changes and messed it up then you can go back to github and download the same file and go back to the good version.**
* **History**- is used to look at files you recently uploaded, you can do this by pressing history.

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.

We have not used history in the class so far to check at the files we have recently uploaded.

**Programming Errors**

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

A character or string incorrectly placed in a command or instruction that causes a failure in execution.

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.

// the setup function runs once when you press reset or power the board

void setup() {

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

pinMode(LED\_BUILTIN, OUTPUT);

}

voud loop()

// the loop function runs over and over again forever

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

}

**The mistake is that there is no bracket to start the void loop. Also void loop is spelled wrong. Also, there are no semi colons after some codes.**

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

A runtime error is a program error that occurs while the program is running. For example, A Logic Error.

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.
2. void setup() {

int ledPIN=12;

pinMode(LED\_BUILTIN, OUTPUT);

Serial begin(9600):

}

I wrote “**ledPin**” instead of “**times**”. The problem is I want to blink the LED 5 times. Therefore, in order to do that instead of **ledPIN** I must write **times** before the Void Setup to symbolize that this is the global variable for a number of times to blink the LED.

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

A logic error is a bug in a program that causes it to operate incorrectly, but it does not crash.

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.
2. int led = 12;

void setup() {

The mistake is that pinMode(led, Output); was not written after “void setup() {. This mistake does not allow the LED does not turn on. So therefore I have to write pinMODE(led, Output) after the void setup in order for the code to work.