|  |  |
| --- | --- |
|  |  |
| **Unit:** Methodology | **Turn In List:** **1. Terms, 2. Post timeline, and 3. Grid** |
| *“I can create and use many data types in a simple computer program.”* | |

**Data Types and Variables: A look at the major data types for modern languages**

**Content Objectives:** Students will be able to declare, initialize and assign variable for a program.

|  |
| --- |
| **Starter Activity** |

// Devin Killinger

//09/19/18

float grade = random(0, 101);

println(grade);

if (grade >= 94) {

println(grade + " Assign letter grade A.");

} else if (grade >= 89 && grade < 94) {

println(grade + " Assign letter grade A-.");

} else if (grade >= 87 && grade < 89) {

println(grade + " Assign letter grade B+.");

} else if (grade >= 83 && grade < 87) {

println(grade + " Assign letter grade B.");

} else if (grade >= 80 && grade < 83) {

println(grade + " Assign letter grade B-.");

} else if (grade >= 77 && grade < 80) {

println(grade + " Assign letter grade C+.");

} else if (grade >= 73 && grade < 77) {

println(grade + " Assign letter grade C.");

} else if (grade >= 70 && grade < 73) {

println(grade + " Assign letter grade C-.");

} else if (grade >= 67 && grade < 70) {

println(grade + " Assign letter grade D+.");

} else if (grade >= 63 && grade < 67) {

println(grade + " Assign letter grade D.");

} else if (grade >= 60 && grade < 63) {

println(grade + " Assign letter grade D-.");

} else {

println(grade + " Assign letter grade F.");

}

|  |  |
| --- | --- |
| **Key Terms:** | |
| Interpreted Language | Reads instructions from a source file without compiling or changing to machine code. |
| Compiled Language | Converts the source files to machine code that is read by the OS |
| Low Level Language | Instruction given to the computer hardware, typically in hex or binary |
| High Level Language | Uses human language terms to give instruction to the compiler |
| Execute | Moving an application to system memory |
| Identifiers | Name given to a data type in memory |
| Declare Variables | Reserving the memory for that data type |
| Initialize Variables | Giving a variable its first value |
| Assign Variables | Change the value in memory using the = command |

|  |
| --- |
| **Assignment:** |
| For each data type give the following information. Use the Processing reference as an aid (note that all data types follow the java standard.) You may write N/A where applicable.   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | **Memory Used** | **Possible Values (Min)** | **Possible Values (Max)** | **Purpose** | **Syntax** | | boolean | 1 bit | 0 or false | 1 or true | Control statements for app flow | boolean a = false; | | byte | 8 bits | -128 | 127 | Sending data | byte b = -128; | | char | 2 bytes  (16 bits) | O | @ | Datatype for characters | char m =’K’; | | color | 32 bits | #000000 | #FFFFFF | Datatype for storing color values | color c2 = #FFCC00 | | double | 64 | More than 3.40282347E+38 | Less than  -3.40282347E+38 | Datatype for floating-point numbers larger than those that can be stored in a float | Double b = -2.984D; | | float | 32 | 3.40282347E+38 | -3.40282347E+38 | Data type for floating-point numbers (e.g. numbers that have a decimal point) | float b = -2.984; | | int | 32 bits | 2,000,000,000 | -199,999,999 | Datatype for integers | int b= -256 | | long | 64 bits | 9,223,372,036,854,775,807 | -9,223,372,036,854,775,808 | Datatype for large integers | Long b = -256; | | String | N/A | alabama | potato | A sequence of characters | -String str1 = “CCCP”  -Println(str1);  -// Prints “CCP” to the console | | XML | N/A | Object.xml | Example.xml | Loads external XML files and create XML objects | loadXML() | | Array | N/A | 1 | 15 | A list of data | datatype[] var  or  var[element] = value  or  var.length | | ArrayList | N/A | squares | ellipses | Stores a variable number of objects | ArrayList<Type>()  Or  ArrayList<Type>(initialCapacity) | | Table | N/A |  |  | Stores data with multiple rows and columns, much like in a traditional spreadsheet | Table()  Or  Table(rows) |   Create a new processing project with a medium gray canvas size of 1000 x 1000 pixels and draw a black grid on the first made up of lines at every 100 pixels vertically and horizontally. Provide text labels (100, 200, etc.) on the left margin and top margin. |

Notes (Points of interest, mistakes, lessons learned, web resources, and thoughts):

|  |
| --- |
|  |