|  |  |
| --- | --- |
| **Date Assigned: 9/22/15** | **Date Due: 9/24/15** |
| **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** |
| // Consider Mr Kapptie’s grading system where numbers  // are turned into letters. Fill in the blanks in the  // following code to complete the boolean expression.  float grade = random(0,100);  if (\_\_\_\_\_\_\_) {  println("Assign letter grade A.");  } else if (\_\_\_\_\_\_\_\_) { // In one conditional statement, you can only ever have one if and one else. However, you can have as many else if's as you like!  println (\_\_\_\_\_\_\_\_);  } else if (\_\_\_\_\_\_\_\_) {  println (\_\_\_\_\_\_\_\_);  } else if (\_\_\_\_\_\_\_\_) {  println (\_\_\_\_\_\_\_\_);  } else {  println (\_\_\_\_\_\_\_\_);  }  // Create a method to use in an app to display letter grade based on the  // position of mouseX on a line. |

|  |  |
| --- | --- |
| **Key Terms:** | |
| Interpreted Language | Most of its implementations execute instructions directly. |
| Compiled Language | Turns source code into machine code. |
| Low Level Language | Closer to machine code. |
| High Level Language | Much more abstraction. Closer to English than machine code. |
| Execute | Perform instruction of program. |
| Identifiers | Name language entities i.e. variable types. |
| Declare Variables | Reserving memory required for datatype. |
| Initialize Variables | Giving its first value at the beginning application. |
| Assign Variables | Assigning a value to a variable. |

|  |
| --- |
| **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 | False | True | Used with control statements to determine the flow of a program. | boolean var  boolean var = booleanvalue | | byte | 8 bits | -128 | 127 | Sends information to and from the serial port and for representing letters in a simpler format. | byte var  byte var = value | | char | 16 bits | 0 | 65,535 | Stores letters and symbols in the Unicode format. | char var  char var = value | | color | 32 bits | 0 | 255 | Stores color values. | color c1 = color(204, 153, 0); | | double | 64 bits | -3.40282347E+38 | 3.40282347E+38 | For floating-point numbers larger than those that can be stored in a float. | double var  double var = value | | float | 32 bits | -3.40282347E+38 | 3.40282347E+38 | Data type for floating-point numbers, e.g. numbers that have a decimal point. | float var  float var = value | | int | 32 bits | -2,147,483,648 | 2,147,483,647 | Datatype for integers, numbers without a decimal point. | int var  int var = value | | long | 64 bits | -9,223,372,036,854,775,808 | 9,223,372,036,854,775,807 | Datatype for large integers. | long var  long var = value | | String | 1 byte per character | N/A | N/A | A string is a sequence of characters. | String(data)  String(data, offset, length) | | XML | Size of file | N/A | N/A | XML is a representation of an XML object, able to parse XML code. | XML(name) | | Array | Depends on data type | N/A | N/A | An array is a list of data. It is possible to have an array of any datatype. | datatype[] var  var[element] = value  var.length | | ArrayList | Depends on data type | N/A | N/A | An ArrayList stores a variable number of objects. | ArrayList<Type>()  ArrayList<Type>(initialCapacity) | | Table | Depends on file size | N/A | N/A | Table objects store data with multiple rows and columns. | Table()  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):

|  |
| --- |
|  |