Unit 2 Lesson 2 - Program Design 1

|  |  |
| --- | --- |
| 1. Program design 2. Code order 3. Use of variables in draw commands | Syntax : int, float, boolean, true, false, = (assign), width, height  Functions (methods) and modularity of design |

1. **Program design**
2. **setup() method** is ran first when the code is executed and runs only once

important code to place in it - *size(), background(), any global variables*

2. Followed by other custom designed functions(methods) – which can be called in setup() and will run (execute) in the order in which they are called.

1. Program modularity
2. Use of various types of comments – top, function and line

|  |  |
| --- | --- |
| Code in Processing can be written  in sequence such as:  size(800,500);  background(30,100,200);  rect(200,200,300,100);  rect (100, 100, 200, 100);  Most programs, however have hundreds if not thousands of lines of code. Written in sequence the code becomes difficult to understand, trace and debug or even change.  This is one of the reasons to use custom designed functions (methods). | **Modularity of design:**  Modularity means that the code is placed in various functions designed to accomplish different tasks .  Then each and every function can be called in setup() which executes once when the program starts.  void setup(){  size(800,500);  background(30,100,200);  design();  }  void design(){  rect (200, 200, 300, 100);  rect (100, 100, 200, 100);  } |

**Program design and use of variables in draw commands.**

1. **Type the following code in Processing. Do NOT include line numbers.**

|  |
| --- |
| 1. /\* Dec 7,2020 2. Student name 3. Variable practice 4. Practice use of variables in design \*/ 5. void setup(){ 6. size(800,500); 7. background(30,100,200); 8. design(); 9. designLines(); 10. } 11. void design(){ 12. rect (200, 200, 300, 100); 13. rect (100, 100, 200, 100); 14. int grayVal = 153; 15. fill(grayVal); 16. rect(10, 10, 55, 55); // Draw gray rectangle 17. grayVal = grayVal + 102; // Assign 255 to grayVal 18. fill(grayVal); 19. rect(35, 30, 55, 55); // Draw white rectangle 20. } 21. void designLines(){ 22. int a = 30; //declare an int variable a and assign 30 to it 23. int b = 40; //declare an int variable b and assign 40 to it 24. line(a, 0, a, height); // height is a Processing variable – holds the height of the screen 25. line(b, 0, b, height); // line formula with use of variables 26. strokeWeight(4); // A calculation can be used as an input to a function 27. line(b-a, 0, b-a, height); 28. } |

1. In your own words explain what effect does the use of variables have on the code?

Variables make the code a little more streamlined and it will also become easier for you to

change multiple things at once while keeping them in proportion to each other. It will also make it far

easier to do things like animate later on.

1. Task1: Create a method which will draw three DIFFERENT, NON -OVERLAPPING ellipses. Use ONLY ONE variable to set the ***position and size*** of the ellipses.
2. Task 2: create a method which will draw series of lines with increasing space between each. Use variables multiplication to create the design.

**The tasks are assigned as homework and evaluated for completion only.**