|  |  |
| --- | --- |
| **Date Assigned: 9/30/16** | **Date Due: 10/4/16** |
| **Unit:** Methodology | **Turn In List:** **1. Terms** |
| *“I will vow to format code so that it is readable and easy to interpret. Good developers don’t try to hide things in source code.”* | |

**Conditions and Formatting Code: Using proper format while introducing conditions in code**

**Content Objectives:** Students will be able to identify and format code appropriately while using appropriate methods with return values.

|  |
| --- |
| **Starter Activity** |
| Modify the Etch-A-Sketch program to respond to keyboard interaction using the following:  int x, y;  void setup() {  size(400,400);  frameRate(10);  // Set start coords  x = 0;  y = 0;  }  void draw() {  fill(255);  //drawName();  //noLoop();  }  // Algorithm for your first name  void drawName() {  moveRight(10);  moveDown(10);  moveLeft(5);  moveUpright(5);  moveDown(5);  moveRight(10);  moveUpright(10);  moveDown(5);  moveDiagonal(5);  moveDia2(5);  moveDia3(5);  moveRight(3);  moveDown(10);  moveUpright(10);  moveDiagonal(10);  moveUpright(10);  println(keyCode);  }  // Method to draw right line  void moveRight(int rep) {  for(int i=0;i<rep\*10;i++){  point(x+i,y);  }  x=x+(10\*rep);  }  // Method to draw line down  void moveDown(int rep) {  for(int i=0;i<rep\*10;i++){  point(x,y+i);  }  y=y+(10\*rep);  }  // Method to draw right line  void moveRight2(int rep) {  for(int i=0;i<rep\*10;i++){  point(x+i,y);  }  x=x+(10\*rep);  }  // Method to draw left line  void moveLeft(int rep) {  for(int i=0;i<rep\*10;i++){  point(x-i,y);  }  x=x-(10\*rep);  }  // Method to draw Uprightline  void moveUpright(int rep) {  for(int i=0;i<rep\*10;i++){  point(x,y-i);  }  y=y-(10\*rep);  }  // Method to draw diagonal line  void moveDiagonal(int rep) {  for(int i=0;i<rep\*10;i++){  point(x+i,y+i);  }  y=y+(10\*rep);  x=x+(10\*rep);    }  // Method to draw dia line  void moveDia2(int rep) {  for(int i=0;i<rep\*10;i++){  point(x-i,y-i);  }  y=y-(10\*rep);  x=x-(10\*rep);  }  // Method to draw dia line  void moveDia3(int rep) {  for(int i=0;i<rep\*10;i++){  point(x+i,y-i);  }  y=y-(10\*rep);  x=x+(10\*rep);  }  void keyPressed() {  if (key == CODED) {  println(keyCode); {  if (keyCode == RIGHT){  moveRight(1);  }else if(keyCode==DOWN){  moveDown(1);  }else if(keyCode==LEFT){  moveLeft(1);  }else if(keyCode==UP){  moveUpright(1);    }  }  }  } |
| Students will save 3 images using the following code:  void mouseClicked() {  saveFrame("line-######.png");  } |

|  |  |
| --- | --- |
| **Key Terms:** | |
| White Space | The space around the sketch |
| Camel or Pascal Case | Words created by concatenating capitalized words |
| Condition | Provides a mean for one thread to suspend execution |
| If | Used to specify a block of code to be executed if condition is true or false |
| If else | Use to specify a bew condition |
| Boolean Expression | Logical statement that is true or false |
| Boolean Variable | Data type having true or false values |

|  |
| --- |
| **Assignment:** |
| Students will explore methods with a return type. Consider the following:  C = (F – 32) \* (5 / 9)  \_\_\_\_\_\_ tempConverter(float \_\_\_\_\_\_\_\_) {  \_\_\_\_\_ \_\_\_\_\_\_\_ = \_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  }  Answer: void setup(){  size(950,300);  background(145);  }  void draw(){  changeToFar(mouseX-100);  fill(50,100,255);  rect(45,50,mouseX-50,20);  fill(255);  drawReference(100,50);  }  void drawReference(int xpos,int ypos){  fill(255);  stroke(200);  line(xpos-50,ypos,900,ypos);  for(int i=0; i<800;i=i+50){  line(xpos+i,ypos-5,xpos+i,ypos+5);  text(i,xpos+i-5,ypos-8);  }    }  float changeToFar(float value){  value= (value)\*5/9;  background(0);  fill(255);  text("Farenheit value="+ (mouseX-100),50,100);  text("Celsius vaule=" + (value),50,130);  return value;  } |
| For this assignment students will create a conversion app that utilizes a method with a return value and the position of the mouse or a line on the screen controlled by the keyboard (or both). Make sure to include the following:   * Title and developer info (your name) * Onscreen instructions * Reference line or shape * Numbered increments and tic marks on screen (hint: use loop) * Updated total as the mouse moves or the arrow keys are pressed   Appropriate conversions may include any of the following:   * Any distance measurement i.e. miles to km etc. * Any volume measurement * Any currency conversion * Math functions i.e. squares or squareroots * Etc. |

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

|  |
| --- |
|  |