# Practical Test 1 Resit

## Instructions

1. This test is covers classes and object composition.
2. You will need the Processing projects PractTest1\_Resit\_PartA and PractTest1\_Resit\_PartB for this test.
3. You should save your work regularly as you progress through the test.
4. Ensure that both the sketch window and the console output is clearly visible when you run the program.
5. At the end of the test, submit your modified versions of *both* Processing projects.

## Help for Part A

In Part A you must use some of Processing’s text functions to display characters of specific sizes at different locations in the sketch. Here is some example code showing how to draw a character ‘X’ at position *(x,y)* with a font size of 100 and colour #67e377:

float x=200, y=300;

float requiredSize = 100;

textSize(requiredSize);

fill(#67e377);

textAlign(CENTER, CENTER);

text('X', x, y);

Note that the command textAlign(CENTER,CENTER) does not always perfectly center characters; sometimes it is a little off, but you can compensate for it by adjusting the *(x,y)* coordinates passed to text() a little bit, e.g.:

text('X', x, y-10);

## Help for Part B

For Part B, you need to generate a random character. This line of code will do that:

char randomLetter = (char)('A'+(int)random(26));

For Part B you also need to arrange letters on a grid. In the example given in the question, the size of the circles is 100 pixels (i.e. radius is 50 pixels), the spacing between them is 150 pixels, and the first letter at the top left of the sketch is centered on (75,75).Part A

Open the project PractTest1\_Resit\_PartA.

The aim of Part A of the test is to create a class called Letter that will draw a letter inside a circle.

The properties of the Letter class are:

* The character to display, e.g. ‘A’, ‘X’, ‘Q’
* An (*x,y*) position on the screen, stored in a PVector
* A colour
* A size

The Letter class should also have a draw() method so that Letter objects can be drawn.

The project already code for the setup() and draw() methods done for you. Here is the code already provided:

Letter l1,l2,l3;

void setup() {

size(600,600);

l1 = new Letter('A',100,100,#3ea567);

l2 = new Letter('B',200,150,#a51289);

l3 = new Letter('Z',500,450,#4c1C98);

}

void draw() {

background(255);

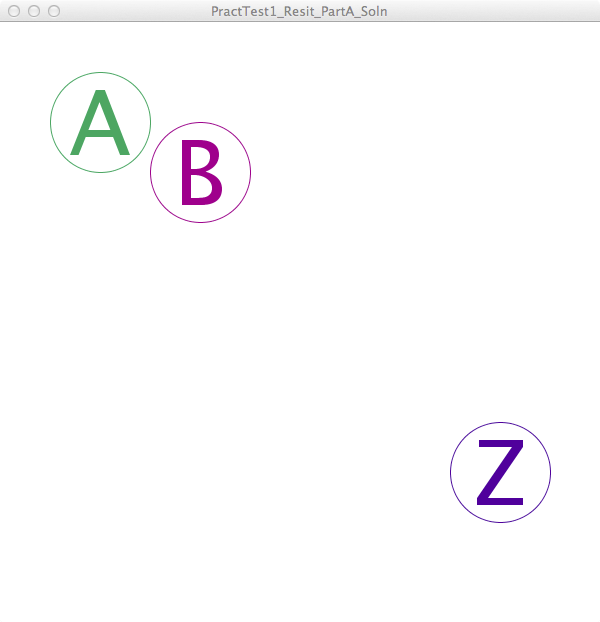
l1.draw();

l2.draw();

l3.draw();

}

When you have completed the Letter class, you should get the following output:

**

Part B

Open the project PractTest1\_Resit\_PartB.

The aim of Part B of the test is to create a class called Grid that will draw many objects of class Letter onto the screen.

You should first of all copy your Letter class from your Part A solution to your Part B solution, as you will need to reuse this class.

Now create a Grid class. The Grid class should have:

* an arraylist of Letter objects which is initialized in the constructor
* an addRandomLetters() method that *uses for loops* to create many Letter objects arranged in a grid with random characters and colours
* a draw() method that draws all the letters in the list

The project already has the main tab’s setup() and draw() methods done for you. Here is the code already provided:

Grid grid;

void setup() {

size(600,600);

grid = new Grid();

grid.addRandomLetters();

}

void draw() {

background(255);

grid.draw();

}

When you have completed the Grid class and included your Letter class from Part A, you should get something like the following output (remember that the colours and letters should be random!):

**Marking Scheme

Note: marks will only be given for programs that compile and run without compilation errors.

|  |  |  |
| --- | --- | --- |
| **Part** | **Item** | **Marks** |
| **A** | Letter has position property | /1 |
|  | Letter has size property | /1 |
|  | Letter has colour property | /1 |
|  | Letter has proper constructor | /2 |
|  | Letter has correct draw() method | /2 |
| **B** | Grid has a properly initialized list of Letter | /2 |
|  | Grid has a correct addRandomLetter() method | /3 |
|  | Grid has a correct draw() method | /3 |
| **TOTAL** |  | **/15** |