

## POO (09:00-10:00)

Write the class **Painter** so that the following code

```
int main()
{
    Painter p(10, 4);
    (((p += 3_right) += 2_down) += 3_right) += 2_up) += 2_left;
    p.print();
    p(0, 3) = 'C';
    p(1, 3) = 'P';
    p(2, 3) = 'P';
    p.print();
    std::cout << "Area = " << (int)p << std::endl;
}
```

compiles and upon execution prints the following to the screen:

```
xxxx.xx...
...X..X...
...xxxx...
.....

xxxx.xx...
...X..X...
...xxxx...
CPP.....

Area = 40
```

Carefully read the main function to deduce what methods/operators should be included in Painter class.

### Constraints:

- You are not allowed to use **STL** at all (for vectors, strings, maps or any template/object defined in STL). The only exception is the usage of “**std::cout**” from the main function
- You are not allowed to use string manipulation functions defined in “string.h” such as **strlen**, **strcpy**, **strdup**, **strtok**, **strcmp**, etc, or string to number conversions.

### Observations:

- The class should allocate a buffer of type char that contains the number of characters required to draw the surface (e.g. for a surface with WIDTH = 5 and HEIGHT = 3 it should allocate 15 characters) all filled with the ‘.’ (point) character.
- The += operator moves an inner cursor to up, down, left or right with a number of characters, filling the space with the character ‘x’

**Grading (informative):**

<b>G1</b>	Destructor	1p
<b>G2</b>	Constructor	2p
<b>G3</b>	Organize your project in 3 files: <b>main.cpp</b> , <b>Painter.h</b> and <b>Painter.cpp</b>	1p
<b>G4</b>	Organize your class <b>Painter</b> to include private and public members, the definition of a constructor, a destructor, and at least one operator. The private data should include one pointer , the width and height of the surface and the cursor (x and y).	2p
<b>G5</b>	Operator += to allow movement in 4 directions (left, right, up and down) leaving a trail of 'x'	9p
<b>G6</b>	Four literals required for the operator+= that specify the direction (1 point per literal)	4p
<b>G7</b>	Operator() to allow access to a character and change it.	4p
<b>G8</b>	Operator to allow the cast to an int that reflects the area (in characters) of the surface.	2p
<b>G9</b>	Function print that prints the content of the surface.	3p
<b>G10</b>	Program runs and works as expected	2p