

C++

Lab_4

Array of objects

1. Create a new project. Add the file *node_coord.h* to the project, in which to place the NODE_COORD class definition:

```
class NODE_COORD
{
    double *pcoord; //pcoord[0] - x, pcoord[1] - y
public:
    NODE_COORD() : pcoord(NULL) {cout << "default constructor: NODE_COORD\n";}
    NODE_COORD(double x, double y); //parameterized constructor
    ~NODE_COORD(); //destructor
    void set(double x, double y); //set x, y coordinates
    void disp(); //display x, y coordinates on screen

private:
    void crash(); //memory allocation error handler - call
                //this function if a memory allocation fails
};
```

Add *node_coord.cpp* and place all class method implementations. The class constructors output to the monitor information about the activity of the constructor, and the destructor - the activity of the destructor.

The main function:

```
int main ()
{
    const double math_pi = 3.141592;
    NODE_COORD tb[3] = { NODE_COORD(0, 0), NODE_COORD(1, 0), NODE_COORD(0, 1) };
    for(size_t i=0; i<3; ++i)
    {
        // display each array element on the monitor
    }

    NODE_COORD * ptr = NULL;
    //Using operator new, dynamically allocate an array of NODE_COORD type for storage
    //of 10 elements and assign its address to ptr pointer.

    for(size_t i=0; i<10; ++i)
    {
        //calculate the x, y coordinates of points uniformly arranged on a circle
        //with radius r0 with the center at the origin of the coordinate system.
        //Each element of the array represents a point lying on this circle.
    }

    fun(ptr); //create a function that will output the coordinates of all points
```

```
        //lying on the circle  
    //release memory for the array tmp.  
    system("pause");  
    return 0;  
}
```