
Lab 10: Dynamic Memory

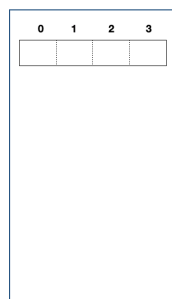
For this lab you do not need to write code but rather will draw pictures to represent what declarations of data look like in memory. You can use any tool you prefer for drawing. I would recommend PowerPoint or Keynote, but you can use any tool such as Google Slides.

You may also draw on a piece of paper, scan it, and turn it in. If choosing this method, make sure your drawings are legible and clean.

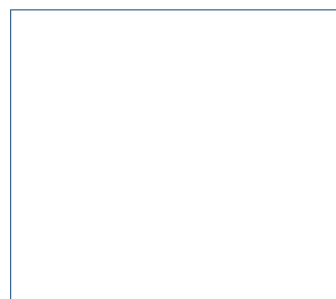
There is no zyBook submission for this lab. Instead, you should submit a pdf file of your answers on **Gradescope**.

Here are three completed examples to give you an idea of what your diagrams should look like.

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     // 1.
7
8     int numbers [4];
9
10    //———— Drawing below —————
11
```



Stack

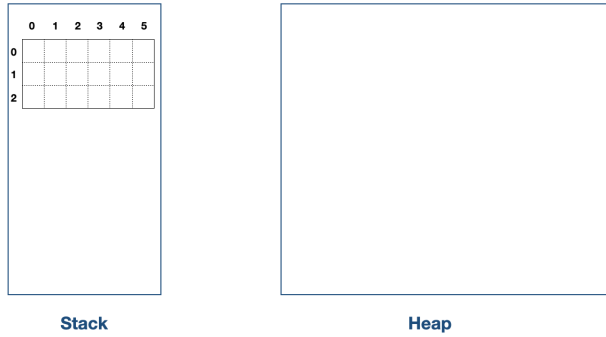


Heap

```
1
2 // 2.
3 char words [3][6];
4
```

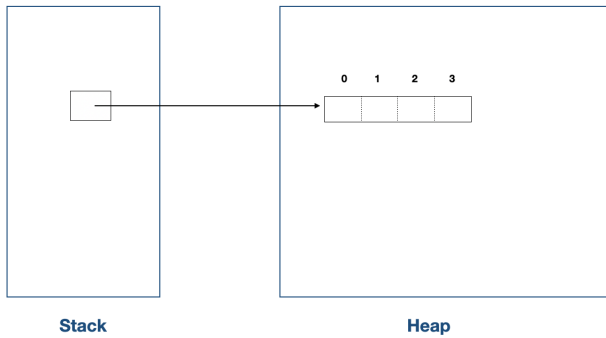
5

//———— Drawing below —————



1
2
3
4
5
6
7
8
9

```
// 3.  
  
int *pNumbers = new int [4];  
  
//———— Drawing below —————  
return 0;  
}
```



Draw the next three cases and submit your solution/drawing as a pdf file on **Gradescope**. Your solutions should clearly show Stack and Heap. Each drawing is 1 point, the last case (6.) is extra point.

```
1 #include<iostream>
2
3 using namespace std;
4
5 int main() {
6
7     // 4.
8
9     char **pWords;
10
11    pWords = new char*[3];
12
13    for( int i=0; i<3; i++) {
14
15        pWords[i] = new char[5];
16
17    }
18
19    //———— Drawing below —————
```

```
1
2     // 5.
3     //Assume we have:
4
5
6     class Date{
7
8         public:
9
10            Date(){
11
12                month = 1;
13                day = 1;
14                year = 200;
15
16            }
17
18        private:
19
20            int month;
21            int day;
22            int year;
23
24
25    };
```

```
26
27
28 // Now draw the following:
29
30 Date weekend [2];
31
32 Date *pDate = new Date [2];
33
34 //———— Drawing below —————
```

```
1
2 //6. Extra Point
3
4 //Assume we have:
5
6
7 struct Node {
8
9     int data;
10
11     Node *pNext;
12
13 };
14
15
16 // Now draw the following:
17
18 Node n1;
19
20 n1.data = 3;
21
22 Node *pNode = new Node;
23
24 pNode->data = 5;
25
26 pNode->pNext = &n1;
27
28
29
30 //———— Drawing below —————
31
32
33 return 0;
34
35 }
```