Ain Shams University
Faculty of Engineering

Software Manual

Assignment 2: Memory Allocation

Names	Mahmoud Hamdy Abd El-Gawwad Mohamed Mohamed El-Morsy
Section	3

Cairo , 2018

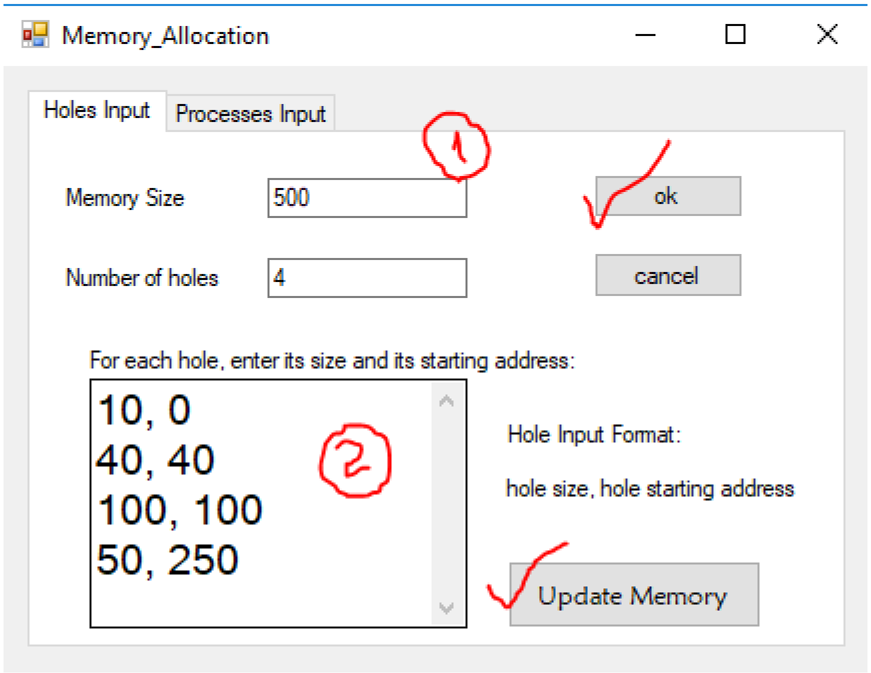
Step 1:

- In “Holes Input” tab, choose Memory Size and Number of Holes, then press the “ok” button.

- Note that you can not change those parameters except in the beginning, so, you can press the ”cancel” button in this tab as much as you want even if you pressed ”update Memory”.

- Enter holes data in the text box below then, press the “Update Memory” button.

- Now, your memory is ready.



Memory Size: 500

Number of holes: 4

For each hole, enter its size and its starting address:

```
10, 0
40, 40
100, 100
50, 250
```

Hole Input Format:

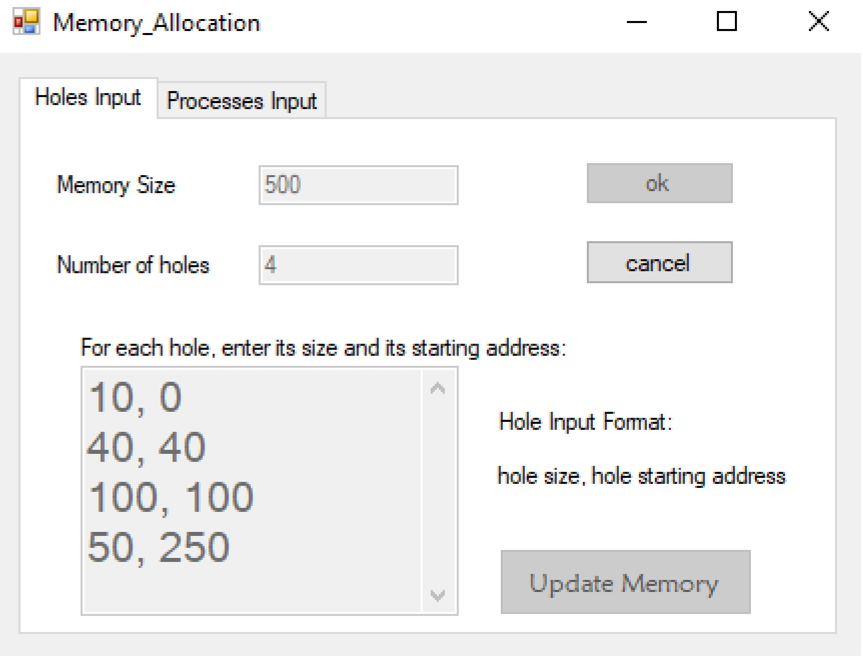
hole size, hole starting address

Step 2:

- In “Processes Input” tab, choose the format to write the processes data depending on whether you are allocating or de-allocating processes.

- In the “Allocation Algorithm” Group Box, check your requested algorithm and press the “Allocate/De-Allocate” button.

- Then, you can view the memory by pressing on “Show Memory” button.



Memory Size 500

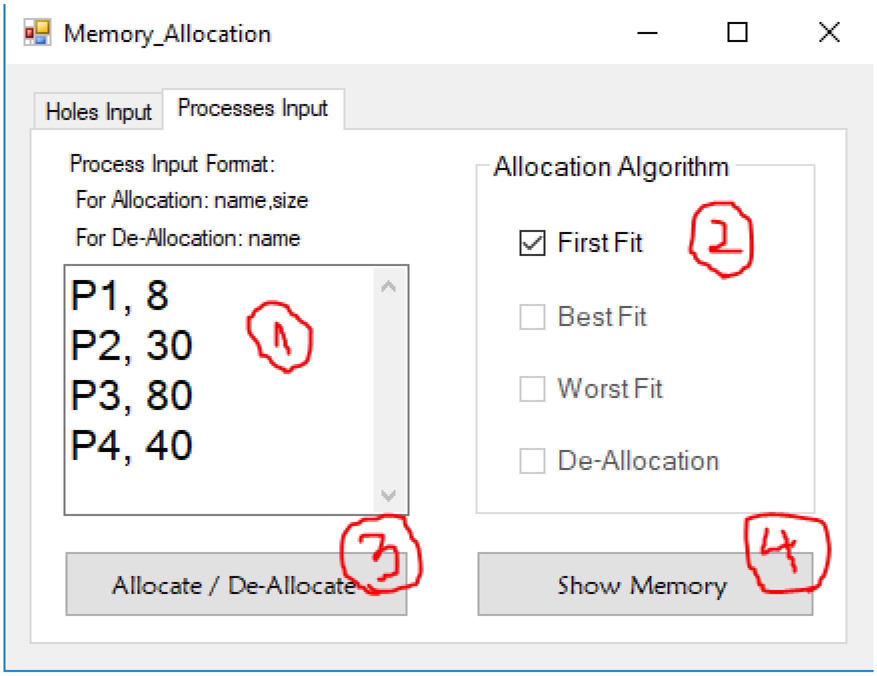
Number of holes 4

For each hole, enter its size and its starting address:

```
10, 0
40, 40
100, 100
50, 250
```

Hole Input Format:

hole size, hole starting address



Process Input Format:
For Allocation: name,size
For De-Allocation: name

```
P1, 8
P2, 30
P3, 80
P4, 40
```

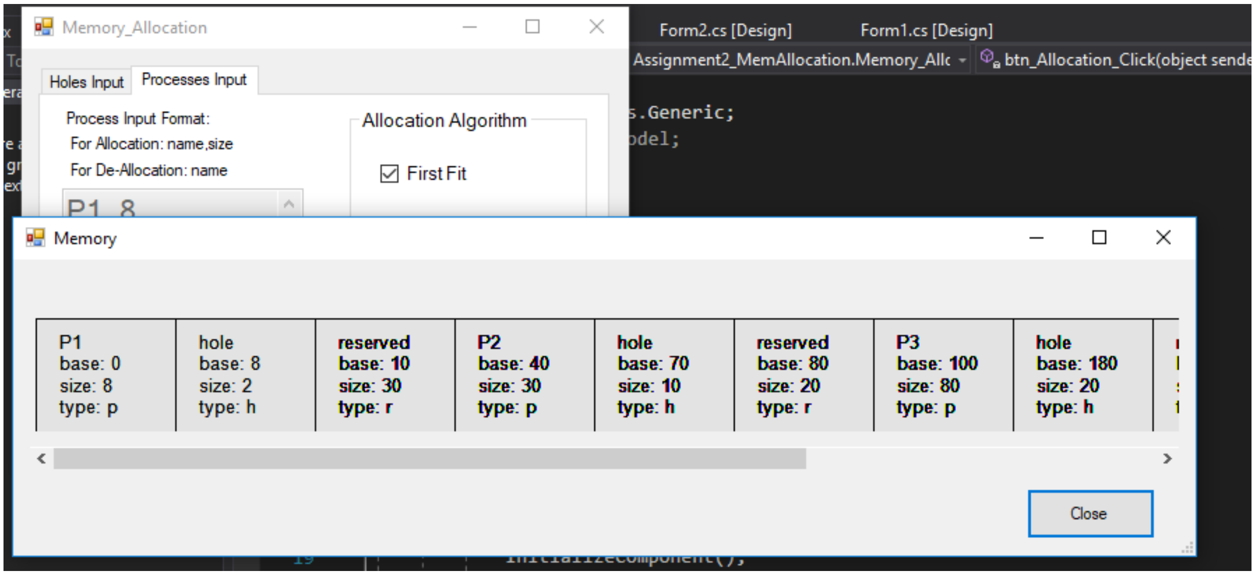
Allocation Algorithm

☑ First Fit

☐ Best Fit

☐ Worst Fit

☐ De-Allocation



Process Input Format:
For Allocation: name,size
For De-Allocation: name

Allocation Algorithm

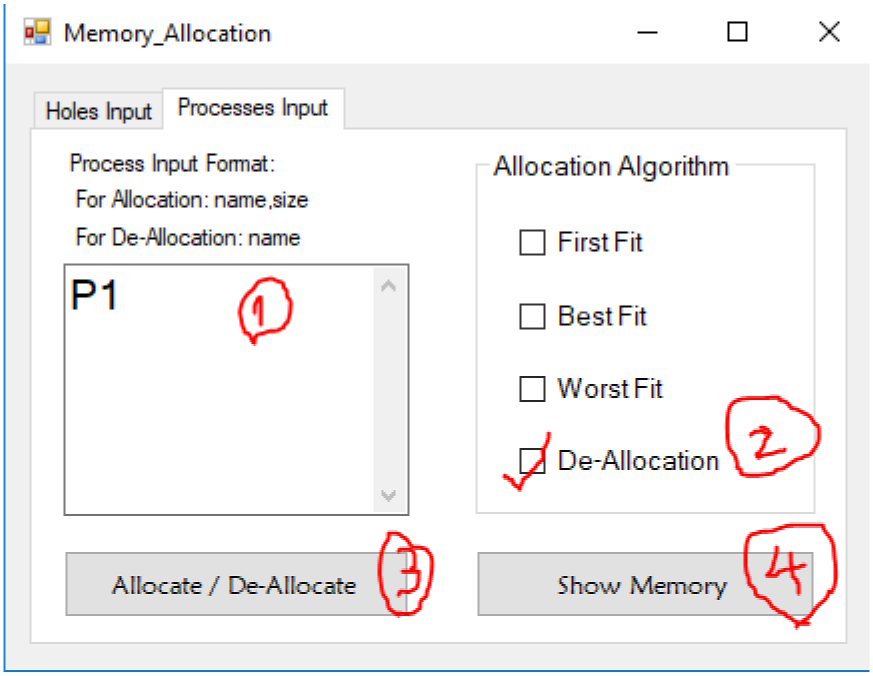
☑ First Fit

```
s.Generic;
odel;
InitializeComponent();
```

P1 base: 0 size: 8 type: p	hole base: 8 size: 2 type: h	reserved base: 10 size: 30 type: r	P2 base: 40 size: 30 type: p	hole base: 70 size: 10 type: h	reserved base: 80 size: 20 type: r	P3 base: 100 size: 80 type: p	hole base: 180 size: 20 type: h

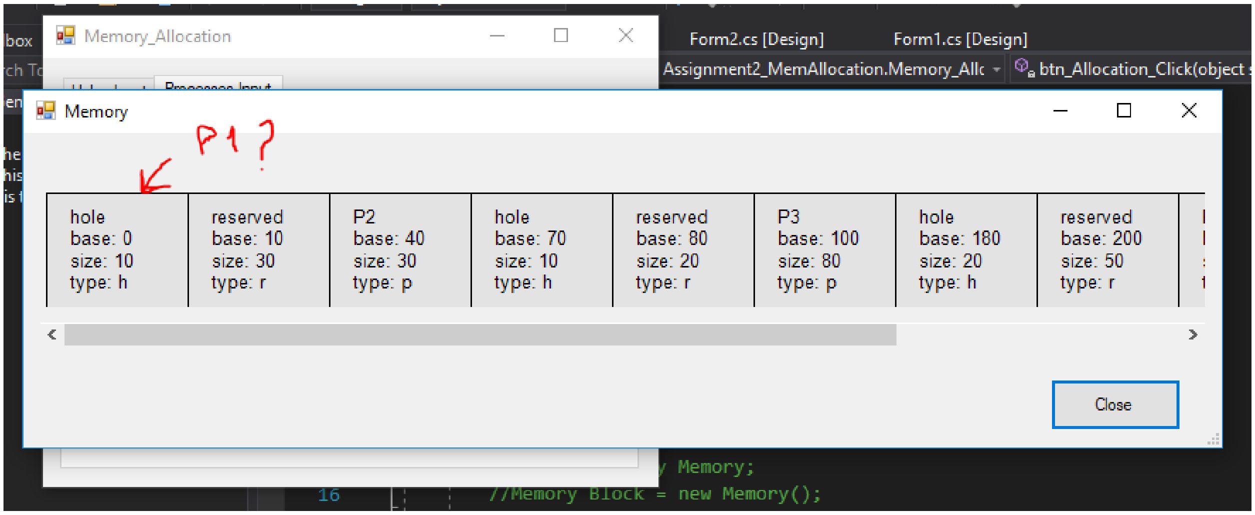
- We can also return again by clicking the “close” button to re-(allocate/De-allocate) a process.

- To De-Allocate a process, write its name, check De-Allocation option, then press ”Allocate/De-Allocate”.



P1

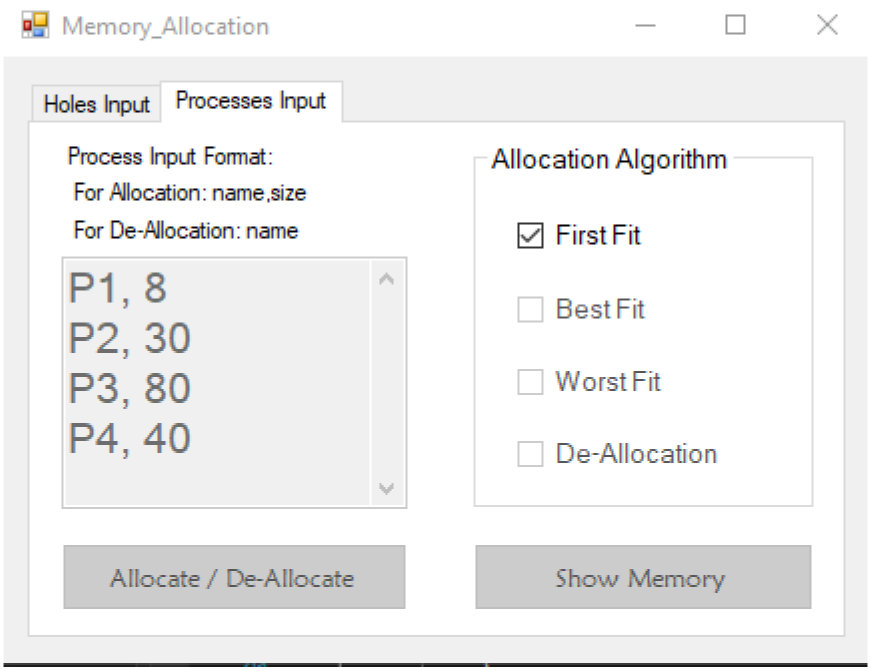
□ First Fit
□ Best Fit
□ Worst Fit
☑ De-Allocation



hole base: 0 size: 10 type: h	reserved base: 10 size: 30 type: r	P2 base: 40 size: 30 type: p	hole base: 70 size: 10 type: h	reserved base: 80 size: 20 type: r	P3 base: 100 size: 80 type: p	hole base: 180 size: 20 type: h	reserved base: 200 size: 50 type: r

```
Memory;
//Memory Block = new Memory();
```

First Fit:



Memory_Allocation

Process Input Format:

For Allocation: name,size

For De-Allocation: name

```
P1, 8
P2, 30
P3, 80
P4, 40
```

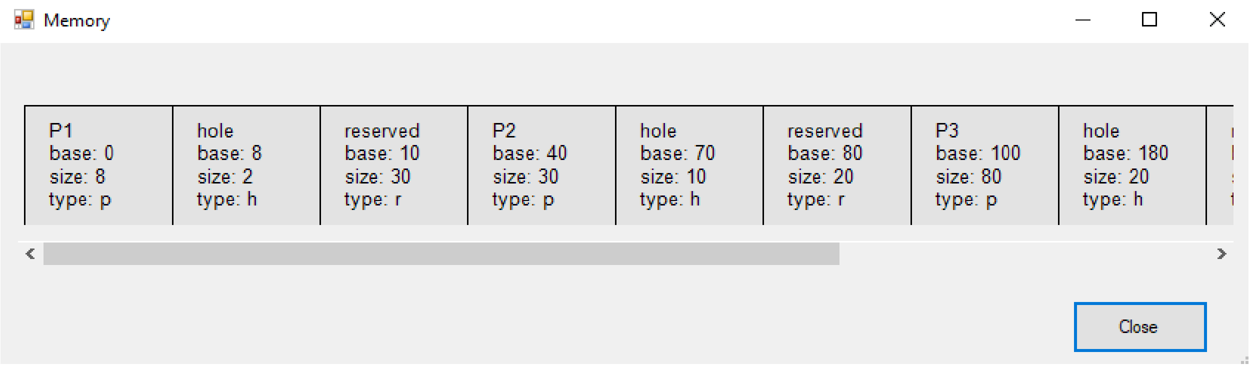
Allocation Algorithm

☑ First Fit

□ Best Fit

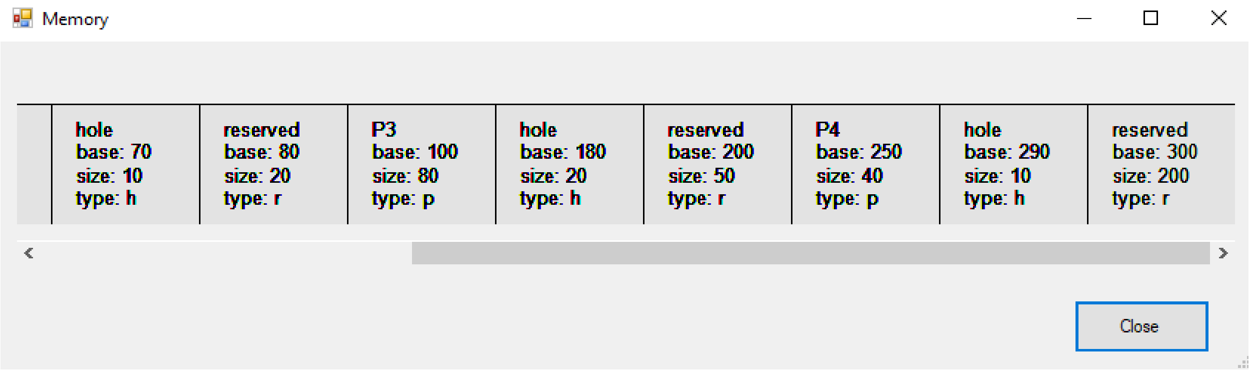
□ Worst Fit

□ De-Allocation



Memory

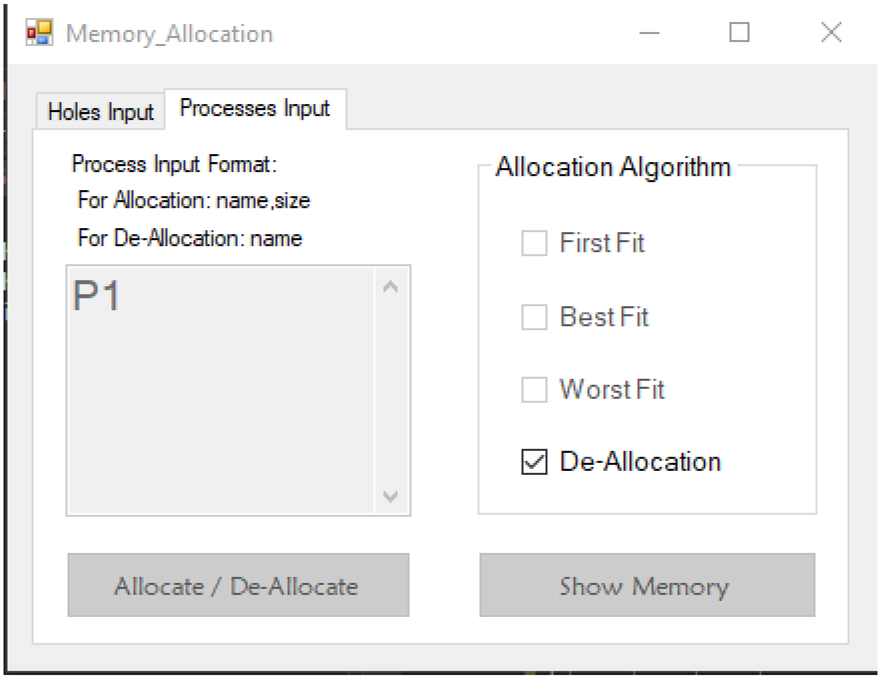
P1 base: 0 size: 8 type: p	hole base: 8 size: 2 type: h	reserved base: 10 size: 30 type: r	P2 base: 40 size: 30 type: p	hole base: 70 size: 10 type: h	reserved base: 80 size: 20 type: r	P3 base: 100 size: 80 type: p	hole base: 180 size: 20 type: h



Memory

hole **base: 70** **size: 10** **type: h**	**reserved** **base: 80** **size: 20** **type: r**	**P3** **base: 100** **size: 80** **type: p**	**hole** **base: 180** **size: 20** **type: h**	**reserved** **base: 200** **size: 50** **type: r**	**P4** **base: 250** **size: 40** **type: p**	**hole** **base: 290** **size: 10** **type: h**	**reserved** **base: 300** **size: 200** **type: r**

De-Allocation:



Process Input Format:
For Allocation: name,size
For De-Allocation: name

P1

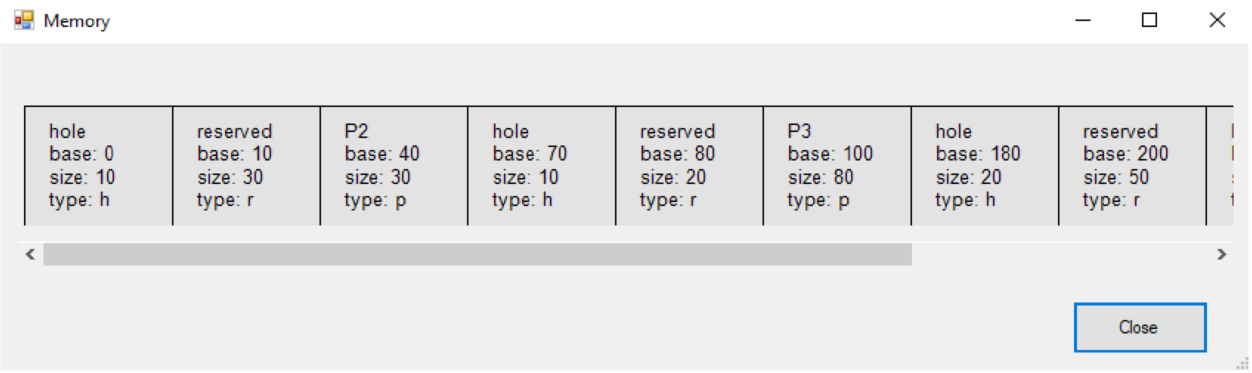
Allocation Algorithm

□ First Fit

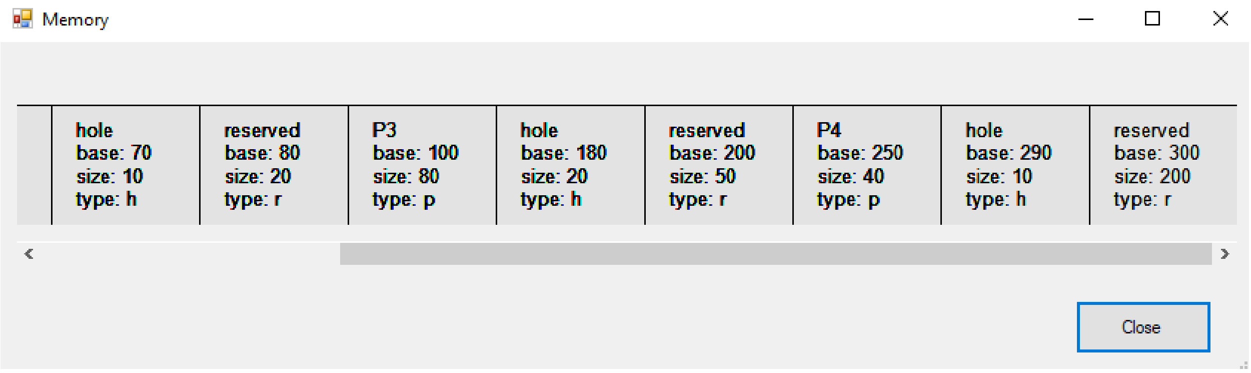
□ Best Fit

□ Worst Fit

☑ De-Allocation

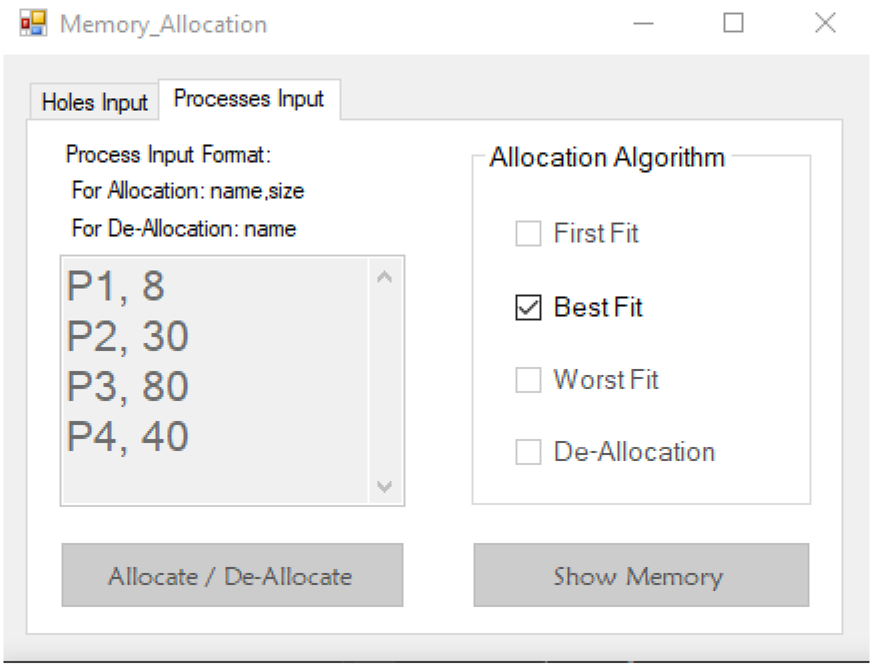


hole base: 0 size: 10 type: h	reserved base: 10 size: 30 type: r	P2 base: 40 size: 30 type: p	hole base: 70 size: 10 type: h	reserved base: 80 size: 20 type: r	P3 base: 100 size: 80 type: p	hole base: 180 size: 20 type: h	reserved base: 200 size: 50 type: r



hole **base: 70** **size: 10** **type: h**	**reserved** **base: 80** **size: 20** **type: r**	**P3** **base: 100** **size: 80** **type: p**	**hole** **base: 180** **size: 20** **type: h**	**reserved** **base: 200** **size: 50** **type: r**	**P4** **base: 250** **size: 40** **type: p**	**hole** **base: 290** **size: 10** **type: h**	**reserved** **base: 300** **size: 200** **type: r**

Best Fit:



Memory_Allocation

Holes Input | Processes Input

Process Input Format:
For Allocation: name,size
For De-Allocation: name

```
P1, 8
P2, 30
P3, 80
P4, 40
```

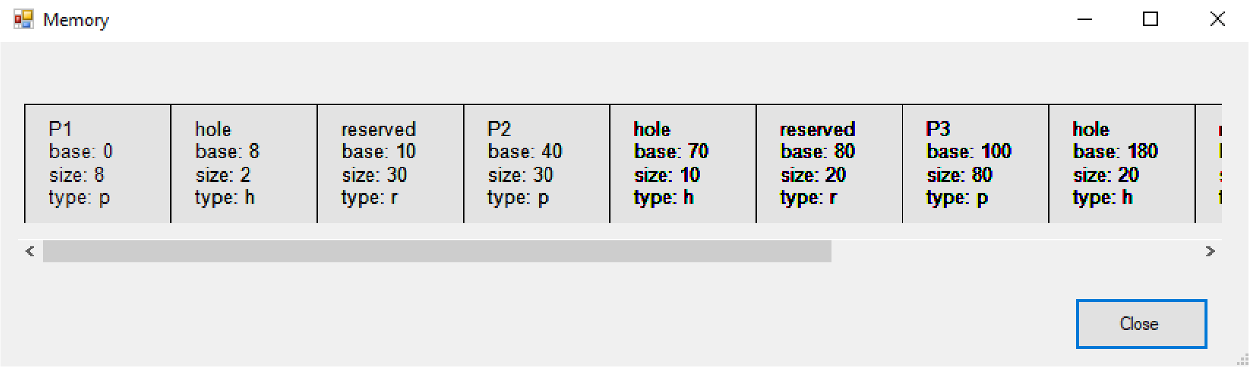
Allocation Algorithm

□ First Fit

☑ Best Fit

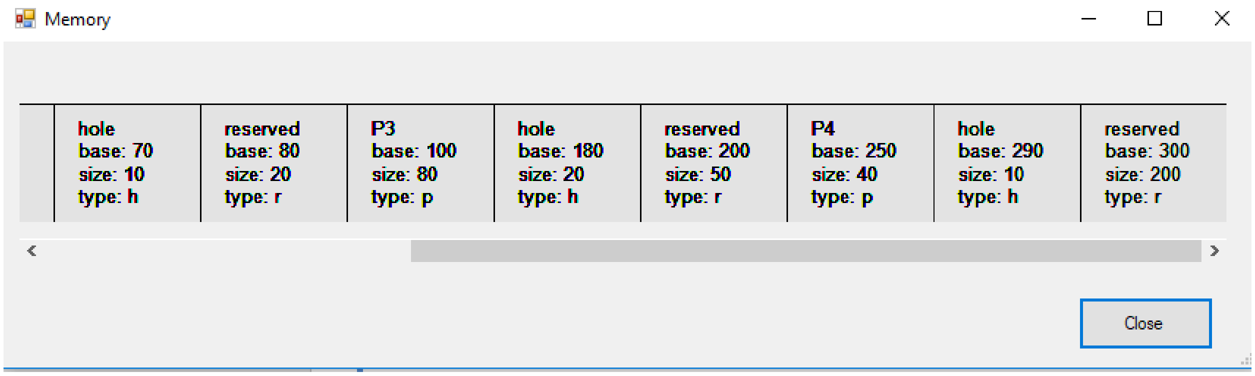
□ Worst Fit

□ De-Allocation



Memory

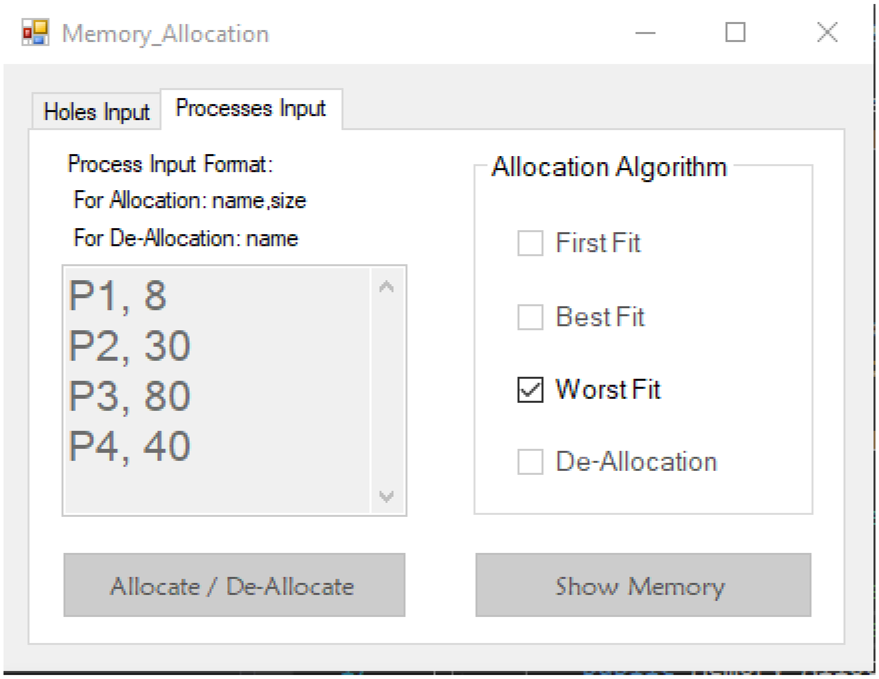
P1 base: 0 size: 8 type: p	hole base: 8 size: 2 type: h	reserved base: 10 size: 30 type: r	P2 base: 40 size: 30 type: p	hole base: 70 size: 10 type: h	reserved base: 80 size: 20 type: r	P3 base: 100 size: 80 type: p	hole base: 180 size: 20 type: h



Memory

hole base: 70 size: 10 type: h	reserved base: 80 size: 20 type: r	P3 base: 100 size: 80 type: p	hole base: 180 size: 20 type: h	reserved base: 200 size: 50 type: r	P4 base: 250 size: 40 type: p	hole base: 290 size: 10 type: h	reserved base: 300 size: 200 type: r

Worst-Fit:



Process Input Format:
For Allocation: name,size
For De-Allocation: name

```
P1, 8
P2, 30
P3, 80
P4, 40
```

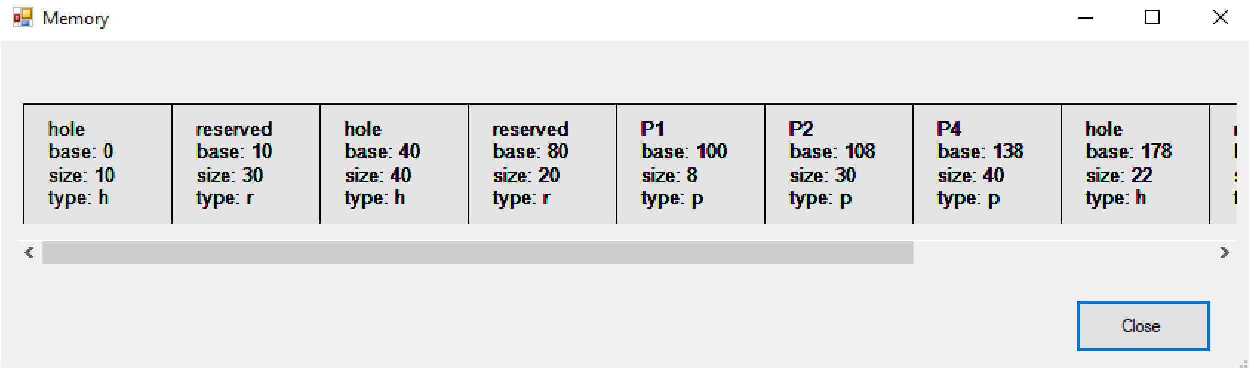
Allocation Algorithm

□ First Fit

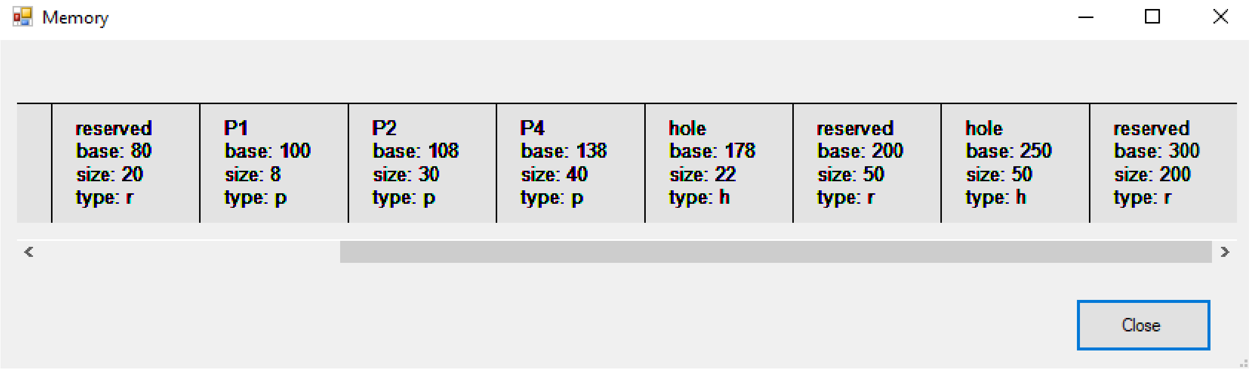
□ Best Fit

☑ Worst Fit

□ De-Allocation



hole base: 0 size: 10 type: h	reserved base: 10 size: 30 type: r	hole base: 40 size: 40 type: h	reserved base: 80 size: 20 type: r	P1 base: 100 size: 8 type: p	P2 base: 108 size: 30 type: p	P4 base: 138 size: 40 type: p	hole base: 178 size: 22 type: h



reserved base: 80 size: 20 type: r	P1 base: 100 size: 8 type: p	P2 base: 108 size: 30 type: p	P4 base: 138 size: 40 type: p	hole base: 178 size: 22 type: h	reserved base: 200 size: 50 type: r	hole base: 250 size: 50 type: h	reserved base: 300 size: 200 type: r