

# **Computer Programming**

Dr. Deepak B Phatak
Dr. Supratik Chakraborty
Department of Computer Science and Engineering
IIT Bombay

Session: Use of Pointers In C++ Programs

## Quick Recap of Relevant Topics



- Basic programming constructs
- Pointer data type in C++
- "Address of" operator in C++
- "Content of" operator in C++
- Caveats when using "address of" and "contents of"

#### Overview of This Lecture

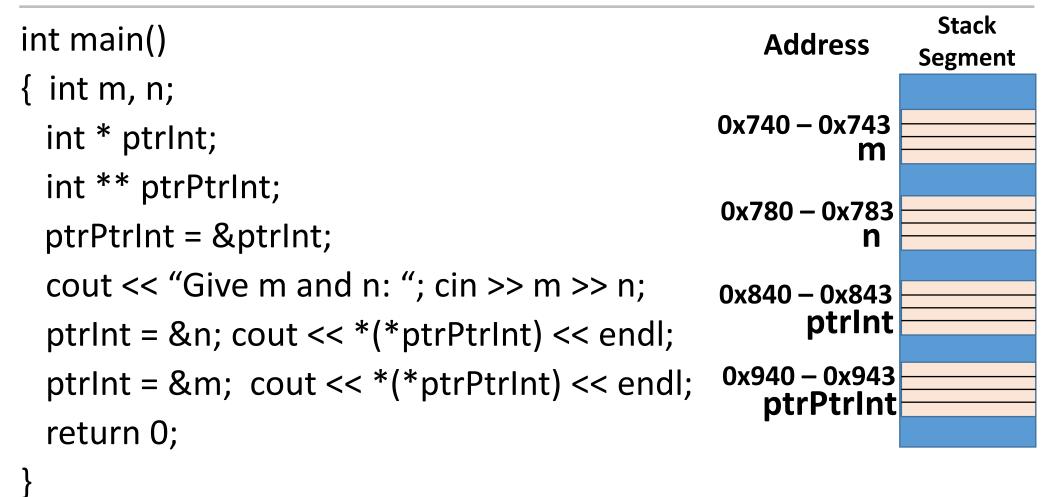


- Understanding usage of pointers in C++ programs
- Understanding how contents of memory locations change when a program with pointers executes



```
int main()
                      If we give input values of 5 and 6
{ int m, n;
               for m and n, what does the program output?
 int * ptrInt;
 int ** ptrPtrInt;
 ptrPtrInt = &ptrInt;
 cout << "Give m and n: "; cin >> m >> n;
 ptrInt = &n; cout << *(*ptrPtrInt) << endl;</pre>
 ptrInt = &m; cout << *(*ptrPtrInt) << endl;</pre>
 return 0;
```







```
Stack
int main()
                                                       Address
                                                                   Segment
{ int m, n;
                                                    0x740 - 0x743
 int * ptrlnt;
                                                              m
 int ** ptrPtrInt;
                                                    0x780 - 0x783
 ptrPtrInt = &ptrInt;
 cout << "Give m and n: "; cin >> m >> n;
                                                    0x840 - 0x843
                                                          ptrInt
 ptrInt = &n; cout << *(*ptrPtrInt) << endl;</pre>
                                                    0x940 - 0x943
 ptrInt = &m; cout << *(*ptrPtrInt) << endl;</pre>
                                                       ptrPtrInt
 return 0;
```



```
Stack
int main()
                                                       Address
                                                                   Segment
{ int m, n;
                                                    0x740 - 0x743
 int * ptrlnt;
 int ** ptrPtrInt;
                                                    0x780 - 0x783
 ptrPtrInt = &ptrInt;
 cout << "Give m and n: "; cin >> m >> n;
                                                    0x840 - 0x843
                                                          ptrInt
 ptrInt = &n; cout << *(*ptrPtrInt) << endl;</pre>
                                                    0x940 - 0x943
 ptrInt = &m; cout << *(*ptrPtrInt) << endl;</pre>
                                                       ptrPtrInt
 return 0;
```



```
Stack
int main()
                                                       Address
                                                                   Segment
{ int m, n;
                                                    0x740 - 0x743
 int * ptrlnt;
 int ** ptrPtrInt;
                                                    0x780 - 0x783
 ptrPtrInt = &ptrInt;
 cout << "Give m and n: "; cin >> m >> n;
                                                    0x840 - 0x843
                                                          ptrInt
 ptrInt = &n; cout << *(*ptrPtrInt) << endl;</pre>
                                                    0x940 - 0x943
 ptrInt = &m; cout << *(*ptrPtrInt) << endl;</pre>
                                                       ptrPtrInt
 return 0;
```



```
Stack
int main()
                                                        Address
                          Content of
                                                                    Segment
{ int m, n;
                    (content of ptrPtrInt)
                                                    0x740 - 0x743
 int * ptrlnt;
                                                                    0x00!
 int ** ptrPtrInt;
                                                    0x780 - 0x783
                                                                    <del>0x006</del>
 ptrPtrInt = &ptrInt;
 cout << "Give m and : "; cin >> m >> n;
                                                    0x840 - 0x843
                                                           ptrInt
 ptrInt = &n; cout << *(*ptrPtrInt) << endl;</pre>
                                                     0x940 - 0x943
 ptrInt = &m; cout << *(*ptrPtrInt) << endl;</pre>
                                                        ptrPtrInt
 return 0;
```



```
Stack
int main()
                                                       Address
                          Content of
                                                                   Segment
{ int m, n;
                           (0x780)
                                                    0x740 - 0x743
 int * ptrlnt;
                                                                   0x00!
 int ** ptrPtrInt;
                                                    0x780 - 0x783
                                                                   0 \times 006
 ptrPtrInt = &ptrInt;
 cout << "Give m and : "; cin >> m >> n;
                                                    0x840 - 0x843
                                                           ptrInt
 ptrInt = &n; cout << *(*ptrPtrInt) << endl;</pre>
                                                    0x940 - 0x943
 ptrInt = &m; cout << *(*ptrPtrInt) << endl;</pre>
                                                       ptrPtrInt
 return 0;
```



```
Stack
int main()
                                                        Address
                                                                    Segment
                            Print 6
{ int m, n;
                                                    0x740 - 0x743
 int * ptrlnt;
                                                                    0x00
 int ** ptrPtrInt;
                                                    0x780 - 0x783
                                                                    0 \times 006
 ptrPtrInt = &ptrInt;
 cout << "Give m and : "; cin >> m >> n;
                                                    0x840 - 0x843
                                                           ptrInt
 ptrInt = &n; cout << *(*ptrPtrInt) << endl;</pre>
 ptrInt = &m; cout << *(*ptrPtrInt) << endl;</pre>
                                                     0x940 - 0x943
                                                        ptrPtrInt
 return 0;
```



```
Stack
int main()
                                                       Address
                                                                   Segment
{ int m, n;
                                                    0x740 - 0x743
 int * ptrlnt;
 int ** ptrPtrInt;
                                                    0x780 - 0x783
 ptrPtrInt = &ptrInt;
 cout << "Give m and n: "; cin >> m >> n;
                                                    0x840 - 0x843
                                                          ptrInt
 ptrInt = &n; cout << *(*ptrPtrInt) << endl;</pre>
 ptrInt = &m; cout << *(*ptrPtrInt) << endl;</pre>
                                                    0x940 - 0x943
                                                       ptrPtrInt
 return 0;
```



```
Stack
int main()
                                                       Address
                         Content of
                                                                  Segment
{ int m, n;
                   (content of ptrPtrInt)
                                                   0x740 - 0x743
 int * ptrlnt;
                                                                   0x00
 int ** ptrPtrInt;
                                                    0x780 - 0x783
                                                                   <del>0x006</del>
 ptrPtrInt = &ptrInt;
 cout << "Give m and
                            "; cin >> m >> n;
                                                   0x840 - 0x843
                                                          ptrInt
 ptrInt = &n; cout << *(\ptrPtrInt) << endl;
 ptrInt = &m; cout << *(*ptrPtrInt) << endl;</pre>
                                                    0x940 - 0x943
                                                       ptrPtrInt
 return 0;
```



```
Stack
int main()
                                                        Address
                          Content of
                                                                    Segment
{ int m, n;
                            (0x740)
                                                    0x740 - 0x743
 int * ptrlnt;
                                                                    0x00
 int ** ptrPtrInt;
                                                     0x780 - 0x783
                                                                    <del>0x006</del>
 ptrPtrInt = &ptrInt;
 cout << "Give m and
                            "; cin >> m >> n;
                                                    0x840 - 0x843
                                                           ptrInt
 ptrInt = &n; cout << *( ptrPtrInt) << endl;</pre>
 ptrInt = &m; cout << *(*ptrPtrInt) << endl;</pre>
                                                     0x940 - 0x943
                                                        ptrPtrInt
 return 0;
```



```
Stack
int main()
                                                        Address
                                                                    Segment
                            Print 5
{ int m, n;
                                                    0x740 - 0x743
 int * ptrlnt;
                                                                    0x00!
 int ** ptrPtrInt;
                                                     0x780 - 0x783
                                                                    0 \times 006
 ptrPtrInt = &ptrInt;
 cout << "Give m and : "; cin >> m >> n;
                                                    0x840 - 0x843
                                                           ptrInt
 ptrInt = &n; cout << *(*ptrPtrInt) << endl;</pre>
 ptrInt = &m; cout << *(*ptrPtrInt) << endl;</pre>
                                                     0x940 - 0x943
                                                        ptrPtrInt
 return 0;
```

## Memory Update using "\*" Operator



- So far, we've used expressions like "\* ptrA" to read the contents of memory at address given by ptrA
- We can also use "\* ptrA" to write the contents of memory at address given by ptrA

\*ptrA = b + c;

stores the value of expression "b + c" as the new content of memory at address given by ptrA



```
int main()
                 If we give values of 5 and 6 for m and n,
{ int m, n, sum =
                     what does the program output?
 int * ptrlnt;
 int * ptrSum;
 ptrSum = ∑
 cout << "Give m and n: "; cin >> m >> n;
 ptrInt = &n; *ptrSum += *ptrInt;
 ptrInt = &m; *ptrSum += *ptrInt;
 cout << "Sum: " << sum << endl;
 return 0;
```



```
Stack
int main()
                                                   Address
                                                              Segment
{ int m, n, sum = 0;
 int * ptrlnt;
                                                0x740 - 0x743
 int * ptrSum;
                                                0x780 - 0x783
 ptrSum = ∑
 cout << "Give m and n: "; cin >> m >> n;
                                                0x7a0 - 0x7a3
                                                       sum
 ptrInt = &n; *ptrSum += *ptrInt;
                                                0x840 - 0x843
 ptrInt = &m; *ptrSum += *ptrInt;
                                                      ptrInt
 cout << "Sum: " << sum << endl;
                                                0x940 - 0x943
                                                    ptrSum
 return 0;
```



```
Stack
int main()
                                                   Address
                                                             Segment
{ int m, n, sum = 0;
 int * ptrlnt;
                                               0x740 - 0x743
 int * ptrSum;
                                                0x780 - 0x783
 ptrSum = ∑
 cout << "Give m and n: "; cin >> m >> n;
                                                0x7a0 - 0x7a3
                                                       sum
 ptrInt = &n; *ptrSum += *ptrInt;
                                                0x840 - 0x843
 ptrInt = &m; *ptrSum += *ptrInt;
                                                      ptrInt
 cout << "Sum: " << sum << endl;
                                                0x940 - 0x943
                                                    ptrSum
 return 0;
```



```
Stack
int main()
                                                   Address
                                                             Segment
{ int m, n, sum = 0;
 int * ptrlnt;
                                               0x740 - 0x743
 int * ptrSum;
                                                0x780 - 0x783
 ptrSum = ∑
 cout << "Give m and n: "; cin >> m >> n;
                                                0x7a0 - 0x7a3
                                                       sum
 ptrInt = &n; *ptrSum += *ptrInt;
                                                0x840 - 0x843
 ptrInt = &m; *ptrSum += *ptrInt;
                                                      ptrInt
 cout << "Sum: " << sum << endl;
                                                0x940 - 0x943
                                                    ptrSum
 return 0;
```



```
Stack
int main()
                                                   Address
                                                             Segment
{ int m, n, sum = 0;
 int * ptrlnt;
                                               0x740 - 0x743
 int * ptrSum;
                                                0x780 - 0x783
 ptrSum = ∑
 cout << "Give m and n: "; cin >> m >> n;
                                                0x7a0 - 0x7a3
                                                       sum
 ptrInt = &n; *ptrSum += *ptrInt;
                                                0x840 - 0x843
 ptrInt = &m; *ptrSum += *ptrInt;
                                                      ptrInt
 cout << "Sum: " << sum << endl;
                                                0x940 - 0x943
                                                    ptrSum
 return 0;
```



```
Stack
int main()
                                                    Address
                      *ptrSum = *ptrSum + *ptrInt
                                                               Segment
{ int m, n, sum = 0;
 int * ptrlnt;
                                                 0x740 - 0x743
                                                               0x00
 int * ptrSum;
                                                 0x780 - 0x783
 ptrSum = ∑
 cout << "Give m and n;"; cin >> m >> n;
                                                 0x7a0 - 0x7a3
                                                               <del>0x000</del>
                                                        sum
 ptrInt = &n; *ptrSum += *ptrInt;
                                                 0x840 - 0x843
 ptrInt = &m; *ptrSum += *ptrInt;
                                                       ptrInt
 cout << "Sum: " << sum << endl;
                                                 0x940 - 0x943
                                                     ptrSum
 return 0;
```



```
Stack
int main()
                                                    Address
                                                               Segment
                         *ptrSum = 0x000 + 0x006
{ int m, n, sum = 0;
 int * ptrlnt;
                                                 0x740 - 0x743
                                                               0x00!
 int * ptrSum;
                                                 0x780 - 0x783
 ptrSum = ∑
 cout << "Give m and n;"; cin >> m >> n;
                                                 0x7a0 - 0x7a3
                                                               <del>0x000</del>
                                                        sum
 ptrInt = &n; *ptrSum += *ptrInt;
                                                 0x840 - 0x843
 ptrInt = &m; *ptrSum += *ptrInt;
                                                       ptrInt
 cout << "Sum: " << sum << endl;
                                                 0x940 - 0x943
                                                     ptrSum
 return 0;
```



```
Stack
int main()
                            *ptrSum = 0x006
                                                    Address
                                                               Segment
\{ int m, n, sum = 0; \}
                       Update contents at address
                             0x7a0 to 0x006
 int * ptrlnt;
                                                   740 - 0x743
                                                               0x00
 int * ptrSum;
                                                 0x780 - 0x783
 ptrSum = ∑
 cout << "Give m and r"; cin >> m >> n;
                                                 0x7a0 - 0x7a3
                                                               0 \times 006
                                                        sum
 ptrInt = &n; *ptrSum += *ptrInt;
                                                 0x840 - 0x843
 ptrInt = &m; *ptrSum += *ptrInt;
                                                       ptrInt
 cout << "Sum: " << sum << endl;
                                                 0x940 - 0x943
                                                     ptrSum
 return 0;
```



```
Stack
int main()
                                                    Address
                                                               Segment
{ int m, n, sum = 0;
 int * ptrlnt;
                                                0x740 - 0x743
 int * ptrSum;
                                                 0x780 - 0x783
 ptrSum = ∑
 cout << "Give m and n: "; cin >> m >> n;
                                                0x7a0 - 0x7a3
                                                        sum
 ptrInt = &n; *ptrSum += *ptrInt;
                                                0x840 - 0x843
 ptrInt = &m; *ptrSum += *ptrInt;
                                                       ptrInt
 cout << "Sum: " << sum << endl;</pre>
                                                 0x940 - 0x943
                                                     ptrSum
 return 0;
```



```
Stack
int main()
                                                    Address
                      *ptrSum = *ptrSum + *ptrInt
                                                               Segment
{ int m, n, sum = 0;
 int * ptrlnt;
                                                0x740 - 0x743
                                                               0x00
 int * ptrSum;
                                                 0x780 - 0x783
 ptrSum = ∑
 cout << "Give m and n:
                           /cin >> m >> n;
                                                0x7a0 - 0x7a3
                                                               0 \times 006
                                                        sum
 ptrInt = &n; *ptrSum += /*ptrInt;
                                                0x840 - 0x843
 ptrInt = &m; *ptrSum += *ptrInt;
                                                       ptrInt
 cout << "Sum: " << sum << endl;
                                                 0x940 - 0x943
                                                     ptrSum
 return 0;
```



```
Stack
int main()
                                                   Address
                                                              Segment
                        *ptrSum = 0x006 + 0x005
{ int m, n, sum = 0;
 int * ptrlnt;
                                                0x740 - 0x743
                                                              0x00
 int * ptrSum;
                                                0x780 - 0x783
 ptrSum = ∑
 cout << "Give m and n: cin >> m >> n;
                                                0x7a0 - 0x7a3
                                                              0 \times 006
                                                       sum
 ptrInt = &n; *ptrSum + *ptrInt;
                                                0x840 - 0x843
 ptrInt = &m; *ptrSum += *ptrInt;
                                                      ptrInt
 cout << "Sum: " << sum << endl;
                                                0x940 - 0x943
                                                    ptrSum
 return 0;
```



```
Stack
int main()
                             *ptrSum = 0x00b
                                                     Address
                                                                Segment
{ int m, n, sum = 0;
                        Update contents at address
                             0x7a0 to 0x00b
 int * ptrlnt;
                                                    740 - 0x743
                                                                0x00
 int * ptrSum;
                                                  0x780 - 0x783
 ptrSum = ∑
 cout << "Give m and n:
                            <mark>√</mark>cin >> m >> n;
                                                 0x7a0 - 0x7a3
                                                                0x00b
                                                         sum
 ptrInt = &n; *ptrSum + */*ptrInt;
                                                 0x840 - 0x843
 ptrInt = &m; *ptrSum += *ptrInt;
                                                        ptrInt
 cout << "Sum: " << sum << endl;
                                                  0x940 - 0x943
                                                      ptrSum
 return 0;
```



```
Stack
int main()
                                                   Address
                                                              Segment
                             Print "Sum: 11"
{ int m, n, sum = 0;
 int * ptrlnt;
                                                0x740 - 0x743
                                                              0x00
 int * ptrSum;
                                                0x780 - 0x783
 ptrSum = ∑
 cout << "Give m and/
                          "; cin >> m >> n;
                                                0x7a0 - 0x7a3
                                                              0x00b
                                                       sum
 ptrInt = &n; *ptrSur /+= *ptrInt;
                                                0x840 - 0x843
 ptrInt = &m; *ptrSam += *ptrInt;
                                                      ptrInt
 cout << "Sum: " << sum << endl;
                                                0x940 - 0x943
                                                    ptrSum
 return 0;
```

#### Summary



- Use of "address of" and "content of" operators in C++
  - Unary & and unary \* operators
- Understanding how contents of memory locations change when executing programs with pointers
- Using "content of" operator to update memory locations