

C++ Primer

Jakub Marel

Revision

Arrays Again

Technicalities

Allocation

Programming in C++ - Primer Lesson 3 - Pointers

Jakub 'Eremiell' Marek <marekj14@fel.cvut.cz>

Silicon Hill C++ Academy

2013/11/04



C++ Primer

Jakub Marel

.

Arrays Agai

Technicalitie

Allocatio

- 1 Revision
- 2 Pointers
- 3 Arrays Again
- 4 Technicalities
- 5 Allocation



Welcome!

C++ Primer

Jakub Marek

Revision

Pointers

Arrays Again

Technicalities

. . .



C++ Primer

Jakub Marek

Revision

Dointor

Arrays Again

Technicalities



C++ Primer

Jakub Marel

Revision

Pointer

Arrays Agair

Tankatanlista

· ccimicantic

Allocation

Conditions



C++ Primer

Jakub Marel

Revision

Pointer

Arrays Again

, ,

rechnicalities

Allocation

Conditions

■ if



C++ Primer

Jakub Marel

Revision

Pointer

Arrays Again

Allays Agail

Technicalitie

Allocatio

Conditions

- if
- switch/case



C++ Primer

Jakub Marel

Revision

ointer

Arrays Agai

Tankai as lista

rechnicalities

Allocatio

Conditions

- if
- switch/case
- ternary



C++ Primer

Jakub Marel

Revision

Pointer

Arrays Agai

T. d. 150

recillicantie

Allocatio

Conditions

- if
- switch/case
- ternary

Loops



C++ Primer

Jakub Mare

Revision

ointer

Arrays Agai

Tankai as lista

MIIOCALIC

Conditions

- if
- switch/case
- ternary

Loops

■ while



C++ Primer

Jakub Mare

Revision

ointer

Arrays Agai

Tankatantista

recimicantic

Allocatio

Conditions

- if
- switch/case
- ternary

Loops

- while
- do/while



C++ Primer

Jakub Mare

Revision

Pointer

Arravs Agai

Technicalitie

Allocatio

Conditions

- if
- switch/case
- ternary

Loops

- while
- do/while
- for



C++ Primer

Jakub Marek

Revision

Arrays Again

Technicalities



C++ Primer

Jakub Marek

Revision

Daintan

Arrays Again

Technicalities

recimeanties

Headers



C++ Primer

Jakub Marek

Revision

Daintar

Arraya Again

+ 1 · 100

recimicantics

Allocation

Headers

 $\quad \blacksquare \ \, {\sf system/library} <>$



C++ Primer

Jakub Marel

Revision

Pointer

. . .

Allays Agaill

Technicalities

Allocation

Headers

- lacktriangleq system/library <>
- local/project ""



C++ Primer

Jakub Marel

Revision

Pointer

Arraya Agair

...-,-...

Technicalities

Headers

- system/library <>
- local/project ""
- shield



C++ Primer

Jakub Marel

Revision

Pointer

Arraya Agair

, 0

Technicalities

Allocation

Headers

- system/library <>
- local/project ""
- shield

Preprocessor directives



C++ Primer

Jakub Marel

Revision

Pointer

Arrays Agair

rechnicalities

Allocatio

Headers

- system/library <>
- local/project ""
- shield

Preprocessor directives

■ include



Headers

C++ Primer

Jakub Marel

Revision

Pointer

Λ...... Λ.....

■ shield

echnicalities

Preprocessor directives

system/library <>local/project ""

- include
- define/undef



C++ Primer

Jakub Marel

Revision

Daintar

Arrays Again

Technicalities

recilineantie.

Allocatio

Headers

- system/library <>
- local/project ""
- shield

Preprocessor directives

- include
- define/undef
- if/ifdef/ifndef/elif/else/endif



C++ Primer

Jakub Marel

Revision

Dointor

. . . .

Allays Agaili

Technicalities

. . .

Headers

- system/library <>
- local/project ""
- shield

Preprocessor directives

- include
- define/undef
- if/ifdef/ifndef/elif/else/endif

I/O Streams



C++ Primer

Jakub Marel

Revision

Daintar

. . .

Allays Agaill

Technicalities

Allocatio

Headers

- system/library <>
- local/project ""
- shield

Preprocessor directives

- include
- define/undef
- if/ifdef/ifndef/elif/else/endif

I/O Streams

■ formatted file input/output



C++ Primer

Jakub Marel

Revision

Pointer

Arrays Agai

. .

Technicalitie

Allocatio

Headers

- system/library <>
- local/project ""
- shield

Preprocessor directives

- include
- define/undef
- if/ifdef/ifndef/elif/else/endif

I/O Streams

- formatted file input/output
- standard in: cin



C++ Primer

Jakub Marel

Revision

Б..

Arrays Again

Technicalities

recillicantie

Allocatio

Headers

- system/library <>
- local/project ""
- shield

Preprocessor directives

- include
- define/undef
- if/ifdef/ifndef/elif/else/endif

I/O Streams

- formatted file input/output
- standard in: cin
- standard out: cout, cerr (error)



C++ Primer

Jakub Marek

Revision

Daintan

Arrays Again

Technicalities

Allocation



C++ Primer

Jakub Marek

Revision

D=:=+==

Arrays Again

Technicalities

Arrays



C++ Primer

Jakub Marel

Revision

Dointor

. . .

Arrays Agair

Technicalities

Allocation

Arrays

■ type name[size]



C++ Primer

Jakub Marel

Revision

Daintan

. . .

rrays Again

Technicalities

Allocation

Arrays

- type name[size]
- fixed size



type name[size]fixed size

C++ Primer

Jakub Marel

Revision

Pointer

Arraya Agair

■ 0 to size - 1

Arrays

Allocation



C++ Primer

Jakub Marel

Revision

Pointer

Arrays Agair

Technicalities

Allocatioi

Arrays

- type name[size]
- fixed size
- 0 to size 1

Variable visibility



C++ Primer

Jakub Marel

Revision

Pointer

Arrays Agair

recnnicalities

Allocatio

Arrays

- type name[size]
- fixed size
- 0 to size 1

Variable visibility

■ global/local



C++ Primer

Jakub Marel

Revision

Pointer

Arraye Agair

■ 0 to s

reclificanties

Allocatio

Arrays

- type name[size]
- fixed size
- 0 to size 1

Variable visibility

- global/local
- block visibility



C++ Primer

Jakub Marel

Revision

Pointer

Arraye Agai

■ 0 to size - 1

echnicalities 0 to Size =

Arrays

Variable visibility

- global/local
- block visibility
- overshadowing

type name[size]fixed size



C++ Primer

Jakub Marel

Revision

Pointer

Arrays Agair

recrimeantie

Allocatic

Arrays

- type name[size]
- fixed size
- 0 to size 1

Variable visibility

- global/local
- block visibility
- overshadowing

Overloading



C++ Primer

Jakub Marel

Revision

Pointer

Arrave Agai

Technicalities

Allocatio

Arrays

- type name[size]
- fixed size
- 0 to size 1

Variable visibility

- global/local
- block visibility
- overshadowing

Overloading

■ same name, different function



Diving deeper

C++ Primer

Jakub Mare

Revision

Pointer

Arrays Agai

■ 0 to

Technicalities ______

Allocatio

Arrays

- type name[size]
- fixed size
- 0 to size 1

Variable visibility

- global/local
- block visibility
- overshadowing

Overloading

- same name, different function
- must have different input types



Diving deeper

C++ Primer

Jakub Mare

Revision

Daintar

arays / igaiii

Technicalities Technicalities

A.II. ...

Arrays

- type name[size]
- fixed size
- 0 to size 1

Variable visibility

- global/local
- block visibility
- overshadowing

Overloading

- same name, different function
- must have different input types
- doesn't care about names





C++ Primer

Jakub Marek

Revision

Pointers

Arrays Again

Technicalities

Allocation



C++ Primer

Jakub Mare

Revision

Pointers

Arrays Agair

Technicalities

r commodificio



C++ Primer

Jakub Mare

Revisio

Pointers

Arrays Agair

Technicalities

. commodnero

Value

■ we get a copy



C++ Primer

Jakub Mare

Revisio

Pointers

Arrays Again

Technicalities

...

Value

- we get a copy
- original remains unchanged



C++ Primer

Pointers

Value

- we get a copy
- original remains unchanged
- we use standard variables



C++ Primer

Jakub Mare

Revision

Pointers

Arrays Agai

Technicalitie

Value

- we get a copy
- original remains unchanged
- we use standard variables

```
int count(int);
int count(int a) {...};
int a;
count(a);
count(10);
```



C++ Primer

Jakub Mare

Revision

Pointers

Arrays Agai

Technicalitie

Allocati

Value Reference

- we get a copy
- original remains unchanged
- we use standard variables

```
int count(int);
int count(int a) {...};
int a;
count(a);
count(10);
```



C++ Primer

Pointers

Value

- we get a copy
- original remains unchanged
- we use standard variables

```
int count(int);
int count(int a) {...};
int a;
count(a);
count (10);
```

Reference

we get the original



C++ Primer

Jakub Mare

Revision

Pointers

Arrays Agair

Technicalitie

Value

- we get a copy
- original remains unchanged
- we use standard variables

Reference

- we get the original
- we can change it

```
int count(int);
int count(int a) {...};
int a;
count(a);
count(10);
```



C++ Primer

Jakub Mare

Revision

Pointers

Arrays Agair

Technicalitie

Value

- we get a copy
- original remains unchanged
- we use standard variables

Reference

- we get the original
- we can change it
- we use pointers

```
int count(int);
int count(int a) {...};
int a;
count(a);
count(10);
```



C++ Primer

Jakub Mare

Revision

Pointers

Arrays Agair

Technicalitie

Value

- we get a copy
- original remains unchanged
- we use standard variables

Reference

- we get the original
- we can change it
- we use pointers



C++ Primer

Jakub Marel

IVENIZION

Pointers

Arrays Again

Technicalities

Allocation



C++ Primer

Jakub Mare

Revisio

Pointers

Arrays Again

Technicalities

A.II. . . .

address of memory, where value is saved



C++ Primer

Jakub Mare

Revisio

Pointers

Arrays Agair

Technicalitie

Alleranden

address of memory, where value is saved is a variable and takes space in memory



C++ Primer

Jakub Mare

Revisio

Pointers

Arrays Agai

Technicalitie

Allocati

address of memory, where value is saved is a variable and takes space in memory is a whole number with constant size of 8/16/32/64b



C++ Primer

Jakub Mare

Revisio

Pointers

Arrays Agai

Technicalitie

Allocati

address of memory, where value is saved is a variable and takes space in memory is a whole number with constant size of 8/16/32/64b each architecture get some maximal memory



C++ Primer

Jakub Mare

Pointers

1 Officers

Technicalities

address of memory, where value is saved is a variable and takes space in memory is a whole number with constant size of 8/16/32/64b each architecture get some maximal memory

2^8	2^16	2^32	2^64
256B	64kB	4GB	16EB=16384PB=16777216GB



C++ Primer

Jakub Marek

Revision

Pointers

Arrays Agair

Technicalities

Allocation



C++ Primer

Jakub Mare

Revisio

Pointers

Δ.....

Technicalitie

- * used as data type means pointer (address)
- * used as operator means dereference (data at address)



C++ Primer

Pointers

- * used as data type means pointer (address)
- * used as operator means dereference (data at address)
- & used as data type means reference (magical address)
- & used as operator means reference (address of data)



* used as data type means pointer (address)

```
C++ Primer
```

Jakub Marel

Revisio

Pointers

Arrays Agair

Technicalitie

```
* used as operator means dereference (data at address)
& used as data type means reference (magical address)
& used as operator means reference (address of data)
#include <iostream>
int main() {
     int a = 10:
     int *b = \&a:
     std::cout << a << " " << *b << std::endl;
     a = 5:
     std::cout << a << " " << *b << std::endl;
     *b = 7:
     std::cout << a << " " << *b << std::endl:
     return 0:
```



C++ Primer

Jakub Marek

Revision

Pointers

Arrays Again

Technicalities

Allocation



C++ Primer

Jakub Marel

Revisio

Pointers

Arrays Agair

Technicalities

...

is a special type of pointer



C++ Primer

Jakub Marel

Revisio

Pointers

Arrays Again

Technicalities

is a special type of pointer is only in C++ (pointers are in C as well)



C++ Primer

Jakub Marel

Revisio

Pointers

Technicalitie

is a special type of pointer is only in C++ (pointers are in C as well) has to be defined at declaration



C++ Primer

Jakub Mare

Revision

Pointers

Arrays Agai

Technicalities

All - - - + : - - -

is a special type of pointer is only in C++ (pointers are in C as well) has to be defined at declaration cannot be changed (the address, not the data)



C++ Primer

Jakub Mare

Revision

Pointers

Arrays Agair

Technicalities

Allocation

is a special type of pointer is only in C++ (pointers are in C as well) has to be defined at declaration cannot be changed (the address, not the data) points to memory, you give it



C++ Primer

Pointers

is a special type of pointer is only in C++ (pointers are in C as well) has to be defined at declaration cannot be changed (the address, not the data) points to memory, you give it behaves like standard variable

```
C++ Primer
```

Jakub Marel

Revisio

Pointers

Arrays Agair

Technicalitie

Allocatio

```
is a special type of pointer
is only in C++ (pointers are in C as well)
has to be defined at declaration
cannot be changed (the address, not the data)
points to memory, you give it
behaves like standard variable
#include <iostream>
int main() {
     int a = 10:
     int \&b = \&a:
     std::cout << a << " " << b << std::endl;
     a = 5:
     std::cout << a << " " << b << std::endl;
     b = 7:
     std::cout << a << " " << b << std::endl;
     return 0:
```



C++ Primer

Jakub Marek

Pointers

Arrays Again

Tochnicalities

All----



C++ Primer

Jakub Marel

Revisio

Pointers

Arrays Agair

Technicalities



C++ Primer

Jakub Mare

Pointers

Arrays Agair

Technicalities

A II = ==+1==

function can only have one return value sometimes we need more



C++ Primer

Jakub Mare

Revision

Pointers

Arraya Aaai

Technicalitie

Allocation

function can only have one return value sometimes we need more make the variable outside of function



C++ Primer

Jakub Mare

Revisio

Pointers

Arraya Agai

Technicalitie

Allocation

function can only have one return value sometimes we need more make the variable outside of function send it in by reference

Return by Parameter

C++ Primer

Jakub Mare

Revision

Pointers

Arrays Agair

Technicalitie

```
function can only have one return value
sometimes we need more
make the variable outside of function
send it in by reference
#include <iostream>
void count(int *a, int &b) {
    *a = *a - b;
    b = *a + 2 * b;
    return;
int main() {
     int a = 5, b = 3;
    std::cout << a << " " << b << std::endl;
    count(&a, b);
    std::cout << a << " " << b << std::endl;
    return 0:
```



Questions?

C++ Primer

Jakub Marek

Pointers

Arrays Again

Technicalities

. . .



C++ Primer

Jakub Marek

_ ...

Pointers

Arrays Again

Technicalities



C++ Primer

Jakub Marel

D . .

Arrays Again

Technicalities

...

arrays are pointers too!



C++ Primer

Jakub Marel

Daintan

Arrays Again

recillicalities

Allocation

arrays are pointers too!
a[] is the same as *a



C++ Primer

Jakub Mare

Dointor

Arrays Again

Technicalities

. . .

arrays are pointers too!

a[] is the same as *a

it just allocates more memory for data



C++ Primer

Jakub Mare

_ .

Pointers

Arrays Again

Technicalities

arrays are pointers too!

a[] is the same as *a

it just allocates more memory for data but you can call it by dereference



C++ Primer

Jakub Mare

Pointers

Arrays Again

Technicalities

...

arrays are pointers too!
a[] is the same as *a
it just allocates more memory for data
but you can call it by dereference
they don't know it's own size!



C++ Primer

Arrays Again

arrays are pointers too! a[] is the same as *a it just allocates more memory for data but you can call it by dereference they don't know it's own size! can be allocated by list

```
C++ Primer
```

Arrays Again

```
arrays are pointers too!
a[] is the same as *a
it just allocates more memory for data
but you can call it by dereference
they don't know it's own size!
can be allocated by list
#include <iostream>
int main() {
     size = 5:
     int a[size];
     for (int i = 0; i < size; i++) {
          *(a + i) = i;
          std::cout << *(a + i) << std::endl;
     return 0;
```



C++ Primer

Jakub Marel

Revision

Pointers

Arrays Again

Technicalities

Allocation



C++ Primer

Jakub Marel

Detaile

Arrays Again

Technicalities

are pointers on pointers



C++ Primer

Jakub Mare

Revision

. -...-

Arrays Again

Technicalitie

are pointers on pointers a [[[]] is the same as **a



C++ Primer

Jakub Mare

revisioi

Pointers

Arrays Again

Technicalitie

are pointers on pointers
a[][] is the same as **a
you always have to name all sizes but the first

4□ > 4□ > 4 = > 4 = > = 9 < 0</p>



C++ Primer

Arrays Again

are pointers on pointers a[][] is the same as **a you always have to name all sizes but the first

as long as you use array notation

4 D > 4 B > 4 B > 4 B > 9 Q P



```
C++ Primer
```

Jakub Marel

Revisio

Pointers

Arrays Again

Technicalities

. ccimicantic

```
#include <iostream>
void printout(int array[][5], int size) {
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < 5; j++) {
            array[i][j] = i * size + j;
            std::cout << array[i][j] << std::endl;
int main() {
    size = 3:
    int a[size][5];
    printout(a, size);
    return 0;
```



```
C++ Primer
```

Jakub Marel

D.

Arrays Again

_

rechnicalities

```
Allocation
```

```
#include <iostream>
void printout(int **array, int sizeX, int sizeY) {
    for (int i = 0; i < sizeX; i++) {
        for (int i = j; j < sizeY; j++) {
            *(*array + i * sizeX + j) = i * sizeX + j;
            std::cout << *(*(array + i) + j) << std::endl;
int main() {
    sizeX = 3:
    sizeY = 5:
    int a[sizeX][sizeY];
    printout(a, sizeX, sizeY);
    return 0:
```



#include <iostream>

return 0:

```
C++ Primer
```

Jakub Marel

Revision

Da:

Arrays Again

Tankainalisiaa

recillicantie

```
void printout(int **array, int sizeX, int sizeY) {
    for (int i = 0; i < sizeX * sizeY; i++) {
        *(*array + i) = i;
        std::cout << *(*array + i) << std::endl;
    }
}
int main() {
    sizeX = 3;
    sizeY = 5;
    int a[sizeX][sizeY];
    printout(a, sizeX, sizeY);</pre>
```



Questions?

C++ Primer

Jakub Marek

Revision

Pointers

Arrays Again

Technicalities

...



C++ Primer

Jakub Marek

Б..

Arrays Again

Technicalities

Allocation



C++ Primer

Jakub Marel

Revision

Folliters

Arrays Agair

Technicalities

you'll get it (sooner or later)



C++ Primer

Jakub Mare

_ .

Arrays Agair

Technicalities

you'll get it (sooner or later)
you touched memory, you haven't allocated



C++ Primer

Jakub Mare

_ .

Arrays Agair

Technicalities

Allocation

you'll get it (sooner or later) you touched memory, you haven't allocated bad pointer



C++ Primer

Jakub Mare

rvevisioi

Fointers

Arrays Again

Technicalities

Allocation

you'll get it (sooner or later) you touched memory, you haven't allocated bad pointer already deleted pointer (NULL them!)



C++ Primer

Jakub Mare

Revision

Arrays Agaii

Technicalities

Allocati

you'll get it (sooner or later)
you touched memory, you haven't allocated
bad pointer
already deleted pointer (NULL them!)
bad array index (it doesn't know it's size!)



C++ Primer

Jakub Marel

Revision

Pointers

Arrays Again

Technicalities

A.II ...



C++ Primer

Jakub Mare

Revision

Pointers

Arrays Again

Technicalities

· ccimicancic

Just as as function can get arguments, your main can get arguments from $\ensuremath{\mathsf{OS}}$



C++ Primer

Technicalities

Just as as function can get arguments, your main can get arguments from OS

./<yourprogram> <argument > . . .



C++ Primer

Technicalities

```
Just as as function can get arguments, your main can get
arguments from OS
```

```
./<yourprogram> <argument > . . .
```

```
int main();
int main(int, char*[]);
```

C++ Primer

Jakub Mare

Revision Pointers Arrays Agair

Technicalities

```
Just as as function can get arguments, your main can get
arguments from OS
./<yourprogram> <argument > . . .
int main();
int main(int, char*[]);
commonly
int main(int argc, char *args[]) {...}
or
int main(int argc, char *argv[]) {...}
```



C++ Primer

Jakub Marek

Revision

Pointers

Arrays Again

Technicalities

...



return 0:

C++ Primer

Jakub Mare

Revision

Arrays Agair

Technicalities

recimicanti

```
#include <iostream>
#include <cstdlib>

int main(int argc, char **argv) {
    int i = strtol(argv[1], NULL, 10);
    int j = strtol(argv[2], NULL, 10);
```

std::cout << i + j << std::endl;



Questions?

C++ Primer

Jakub Marek

rtevision

Pointers

Arrays Again

Technicalities

Allocation



Memory Allocation

C++ Primer

Jakub Marel

Revision

Pointers

Arrays Again

Technicalities

Allocation



Memory Allocation

C++ Primer

Jakub Mare

Pointer

Pointers

Arrays Agai

Technicalities

Allocation

memory can be allocated statically (on stack) or dynamically (on heap)



Memory Allocation

C++ Primer

Jakub Mare

Revision

Pointers

Arrays Agai

Technicalities

Allocation

memory can be allocated statically (on stack) or dynamically (on heap) you already know static allocation

C++ Primer

Jakub Mare

Revision

Pointers

Arrays Agair

Technicalitie

Allocation

memory can be allocated statically (on stack) or dynamically (on heap) you already know static allocation

```
int i = 5;
double d = 6.2;
int a[5] = \{1, 2, 3, 4, 5\};
```

C++ Primer

Jakub Mare

Revision

Pointers

Arrays Agair

Technicalities

Allocation

```
memory can be allocated statically (on stack) or dynamically (on heap) you already know static allocation
```

```
int i = 5;
double d = 6.2;
int a[5] = \{1, 2, 3, 4, 5\};
```

has it's limits

C++ Primer

Jakub Mare

Revision

Arrays Agai

Technicalities

· cciiii caiicic

Allocation

memory can be allocated statically (on stack) or dynamically (on heap) you already know static allocation

int i = 5;
double d = 6.2;

int $a[5] = \{1, 2, 3, 4, 5\};$

has it's limits

you have to know the size on compilation time

C++ Primer

Jakub Mare

Pointers

Arrays Agair

Technicalitie

Allocation

memory can be allocated statically (on stack) or dynamically (on heap) you already know static allocation

```
int i = 5;
double d = 6.2;
int a[5] = \{1, 2, 3, 4, 5\};
```

has it's limits you have to know the size on compilation time size cannot change

C++ Primer

Jakub Mare

Pointers
Arrays Agair

Technicalities

Allocation

memory can be allocated statically (on stack) or dynamically (on heap) you already know static allocation

```
int i = 5;
double d = 6.2;
int a[5] = \{1, 2, 3, 4, 5\};
```

has it's limits you have to know the size on compilation time size cannot change you can't create new objects

C++ Primer

Jakub Mare

Pointers Arrays Agair

Technicalities

Allocation

memory can be allocated statically (on stack) or dynamically (on heap) you already know static allocation

```
int i = 5;
double d = 6.2;
int a[5] = \{1, 2, 3, 4, 5\};
```

has it's limits
you have to know the size on compilation time
size cannot change
you can't create new objects
it's all allocated on program start



C++ Primer

Jakub Marel

Revisio

Fointers

Arrays Again

Technicalities



C++ Primer

Jakub Mare

Revision

Pointers

Arrays Agaii

Technicalities

Allocation

much more flexible



C++ Primer

Jakub Mare

T.CVISIOI

Folliters

Arrays Agair

Technicalities

Allocation

much more flexible danger of memory leak



C++ Primer

Jakub Mare

n . .

A

T. . I.

r commount of

Allocation

much more flexible danger of memory leak need for manual deallocation



C++ Primer

Jakub Mare

T.CVISIOI

Pointers

Arrays Agair

Technicalities

Allocation

much more flexible danger of memory leak need for manual deallocation you always get pointers (or objects)



C++ Primer

Jakub Mare

Trevision

Allays Agail

Technicalities

Allocation

much more flexible danger of memory leak need for manual deallocation you always get pointers (or objects) operators new and delete

C++ Primer

Jakub Mare

Pointer

Arrays Agai

Technicalities

```
much more flexible
danger of memory leak
need for manual deallocation
you always get pointers (or objects)
operators new and delete
int *i = new int(5);
delete i:
int *a = new int[5]{1, 2, 3, 4, 5};
delete[] i;
```



```
C++ Primer
```

Jakub Marel

Revisio

Dointon

Arrays Agai

Technicalities

recillicantie

```
#include <iostream>
#include <cstdlib>
int main(int argc, char **argv) {
    int sizeX = strtol(argv[1], NULL, 10);
    int sizeY = strtol(argv[2], NULL, 10);
    int **a = new int*[sizeX];
    for (int i = 0; i < sizeX; i++) {
        a[i] = new int[sizeY];
        for (int j = 0; j < sizeY; j++) {
            a[i][j] = i * sizeY + j;
            std::cout << a[i][j] << std::endl;
    for (int i = 0; i < sizeY; i++) {
        delete[] a[i];
    delete[] a;
    return 0:
```



Questions?

C++ Primer

Jakub Marek

Revision

Pointers

Arrays Again

Technicalities



Break!

C++ Primer

Jakub Marek

Pointers

Arrays Again

Technicalities