

▲ロト ▲帰ト ▲ヨト ▲ヨト 三日 - の々ぐ

Computer Programming

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

Session: Graphics objects in arrays and structures Guest Lecturer: Dr. Abhiram Ranade

Quick recap



◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?





Last sessions: Graphics facilities in Simplecpp.





Last sessions: Graphics facilities in Simplecpp.

This session:





▲ロト ▲帰ト ▲ヨト ▲ヨト 三日 - の々ぐ

Last sessions: Graphics facilities in Simplecpp.

This session:

Graphics objects in structures/arrays.



▲ロト ▲帰ト ▲ヨト ▲ヨト 三日 - の々ぐ

Last sessions: Graphics facilities in Simplecpp.

This session:

- Graphics objects in structures/arrays.
- A fun application



Last sessions: Graphics facilities in Simplecpp.

This session:

- Graphics objects in structures/arrays.
- A fun application

Reference: Chapters 5, 20 of "An introduction to programming through C++", McGraw Hill Education, 2014.



◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?



▲ロト ▲帰ト ▲ヨト ▲ヨト 三日 - の々ぐ

You can make an array of graphics objects, e.g.



You can make an array of graphics objects, e.g.

Rectangle r[10];





You can make an array of graphics objects, e.g.

```
Rectangle r[10];
```

This creates 10 degenerate rectangles with zero width, height etc.



You can make an array of graphics objects, e.g.

```
Rectangle r[10];
```

This creates 10 degenerate rectangles with zero width, height etc.

You must then make it have the appropriate position, height, width by calling reset, e.g.



You can make an array of graphics objects, e.g.

```
Rectangle r[10];
```

This creates 10 degenerate rectangles with zero width, height etc.

You must then make it have the appropriate position, height, width by calling reset, e.g.

```
r[5].reset(100,100,50,70);
```



You can make an array of graphics objects, e.g.

```
Rectangle r[10];
```

This creates 10 degenerate rectangles with zero width, height etc.

You must then make it have the appropriate position, height, width by calling reset, e.g.

```
r[5].reset(100,100,50,70);
```

Similarly for other graphics objects.



◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?



▲□▶ ▲圖▶ ★ 国▶ ★ 国▶ - 国 - のへで

Can be created in the natural manner, e.g.



▲ロト ▲帰ト ▲ヨト ▲ヨト 三日 - の々ぐ

Can be created in the natural manner, e.g.

```
Rectangle *rptr = new Rectangle(100, 100, 50, 70);
```



▲ロト ▲帰ト ▲ヨト ▲ヨト 三日 - の々ぐ

Can be created in the natural manner, e.g.

```
Rectangle *rptr = new Rectangle(100, 100, 50, 70);
```

The new operator will get called with the required constructor.

Graphics objects in structures



◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?

Graphics objects in structures



Are allowed, e.g.



Graphics objects in structures



Are allowed, e.g.

struct Button{Rectangle border; Text label;};





Are allowed, e.g.

```
struct Button{Rectangle border; Text label;};
```

This will create border and label using default constructors. The correct values can be given using reset.



◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?

► A 4 × 4 board



◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?

- A 4 × 4 board
- 15 tiles of size 1×1 , numbered 1 through 15.



◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 のへぐ

A 4 × 4 board



◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 のへ⊙

15 tiles of size 1 × 1, numbered 1 through 15.
Initially placed randomly in the 16 possible positions.

A 4 × 4 board



15 tiles of size 1 × 1, numbered 1 through 15.
Initially placed randomly in the 16 possible positions.

Goal: Arrange the tiles in order 1..15 top to bottom left to right, i.e. so that it looks like

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

A 4 × 4 board



15 tiles of size 1 × 1, numbered 1 through 15.
Initially placed randomly in the 16 possible positions.

Goal: Arrange the tiles in order 1..15 top to bottom left to right, i.e. so that it looks like

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Operations allowed: If a tile is adjacent to an empty position, it can be slid into the empty position.

A 4 × 4 board



・ロト ・ 同 ト ・ ヨ ト ・ ヨ ・ うへの

15 tiles of size 1 × 1, numbered 1 through 15.
Initially placed randomly in the 16 possible positions.

Goal: Arrange the tiles in order 1..15 top to bottom left to right, i.e. so that it looks like

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Operations allowed: If a tile is adjacent to an empty position, it can be slid into the empty position.

Getting the required order is not always possible!



◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?



► Tile = rectangle and + text giving the number.





▲ロト ▲帰ト ▲ヨト ▲ヨト 三日 - の々ぐ

- ► Tile = rectangle and + text giving the number.
- Board = a two dimensional array slots of type Slot, where:



- ► Tile = rectangle and + text giving the number.
- Board = a two dimensional array slots of type Slot, where: slots[i][j] will hold a pointer to the rectangle in it if any and the text in it if any.



- ► Tile = rectangle and + text giving the number.
- Board = a two dimensional array slots of type Slot, where: slots[i][j] will hold a pointer to the rectangle in it if any and the text in it if any.
- The board will be initialized in some random manner.



- ► Tile = rectangle and + text giving the number.
- Board = a two dimensional array slots of type Slot, where: slots[i][j] will hold a pointer to the rectangle in it if any and the text in it if any.
- The board will be initialized in some random manner.
- When the user clicks, we check if the clicked slot is next to the empty slot. If so, the rectangle and the text in the clicked slot is moved into the empty slot.

Remark regarding the puzzle:



Remark regarding the puzzle:



Celebrated puzzle from 19th century.



Remark regarding the puzzle:



Celebrated puzzle from 19th century.

Theorem: From an arbitrary initial position, you can reach one of the following positions but not both.

1	2	3	4	1	2	3	4
5	6	7	8	5	6	7	8
9	10	11	12	9	10	11	12
13	14	15		13	15	14	



◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?







▲ロト ▲帰ト ▲ヨト ▲ヨト 三日 - の々ぐ

Main goals of the lecture

Study how to create arrays/structures of graphics objects.



- Study how to create arrays/structures of graphics objects.
- Nothing really new was stated!



- Study how to create arrays/structures of graphics objects.
- Nothing really new was stated!
- Give another example of graphics..



- Study how to create arrays/structures of graphics objects.
- Nothing really new was stated!
- Give another example of graphics..
- Arrays of objects will be needed if you want lots of objects on the screen, e.g. gas molecules bouncing around in a chamber.



Main goals of the lecture

- Study how to create arrays/structures of graphics objects.
- Nothing really new was stated!
- Give another example of graphics..
- Arrays of objects will be needed if you want lots of objects on the screen, e.g. gas molecules bouncing around in a chamber.

Chapter 20 gives an example of a "Snake" game that uses arrays of graphics objects.



Main goals of the lecture

- Study how to create arrays/structures of graphics objects.
- Nothing really new was stated!
- Give another example of graphics..
- Arrays of objects will be needed if you want lots of objects on the screen, e.g. gas molecules bouncing around in a chamber.

Chapter 20 gives an example of a "Snake" game that uses arrays of graphics objects.

The possibilities for doing interesting projects are endless..



Main goals of the lecture

- Study how to create arrays/structures of graphics objects.
- Nothing really new was stated!
- Give another example of graphics..
- Arrays of objects will be needed if you want lots of objects on the screen, e.g. gas molecules bouncing around in a chamber.

Chapter 20 gives an example of a "Snake" game that uses arrays of graphics objects.

The possibilities for doing interesting projects are endless..

End of lecture series on graphics.