COMP 250
INTRODUCTION TO COMPUTER SCIENCE

Week 5 1 - Arraylist

Giulia Alberini, Fall 2020

- Arraylist

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

ARRAY LISTS

- **Arrays whose elements have a primitive type**

```
int[] myInts = new int[15];
myInts[3] = -732;
```

- **Arrays whose elements have a reference type**
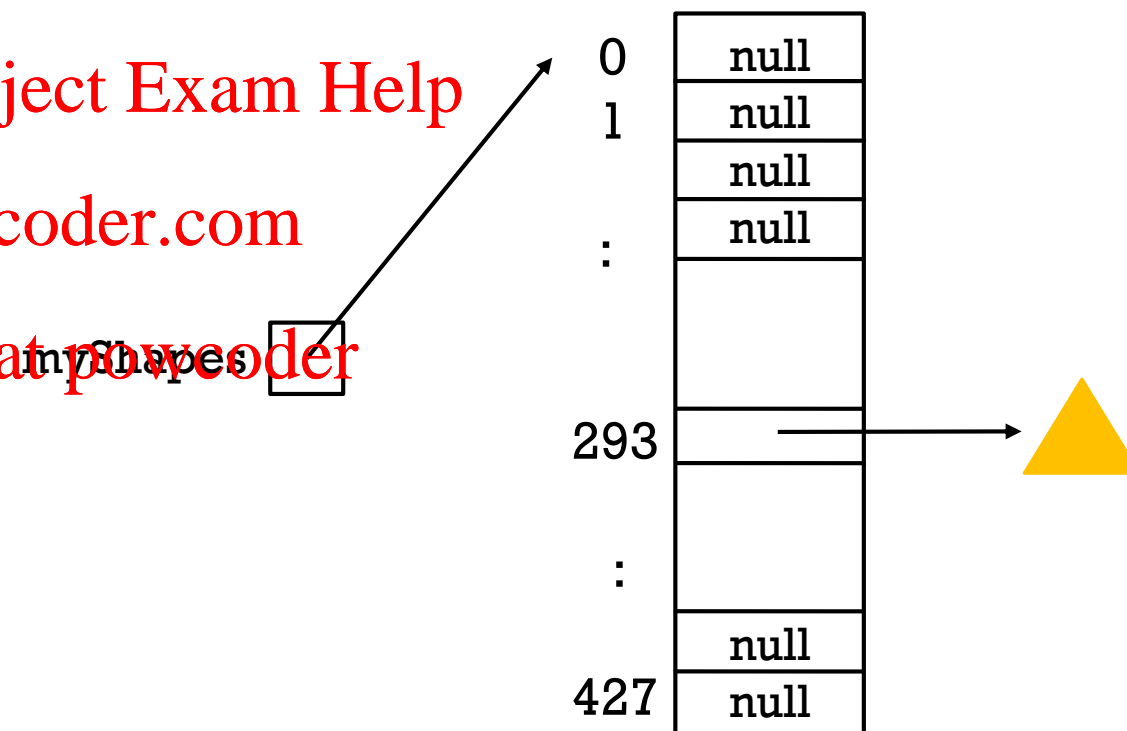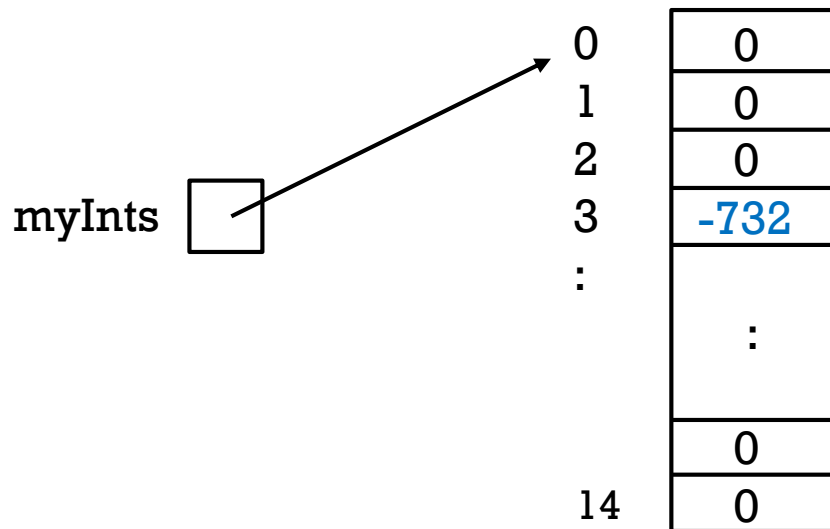
```
Shape[] myShapes = new Shape[428];
shapes[293] = new Shape( ▲ );
```

```
int[] myInts = new int[15];
myInts[3] = -732;
```

```
Shape[] myShapes = new Shape[428];
shapes[293] = new Shape( △ );
```

myInts	
0	0
1	0
2	0
3	-732
:	:
	0
14	0

myShapes

0	null
1	null
	null
	null
:	
293	
:	
	null
427	null

You can think of an array as a block of consecutive slots in memory

A computer accesses an element in an array in constant time

i.e. constant, independent of the length N of the array.

```
    ....   =  a[k] ;          // read
```

```
    a[k]   =   .... ;          // write
```

You will learn more about how this works in COMP 206 and 273.

An ordered set of elements

$$a_0, a_1, a_2, ..., a_{N-1}$$

N is the number of elements in the list, often called the "size" of the list.

WHAT WOULD WE LIKE TO DO WITH A LIST?

get(i)   // Returns the i-th element (but doesn't remove it)

set(i,e)   // Replaces the i-th element with e

Assignment Project Exam Help

add(e)   // Append element e at the end of the list

https://powcoder.com

add(i,e)   // Inserts element e into the i-th position

remove(i)   // Removes the i-th element from list

Add WeChat powcoder

remove(e)   // Removes first occurrence of element e

                // from the list (if it is there)

clear()   // Empties the list.

isEmpty()   // Returns true if empty, false if not empty.

size()   // Returns number of elements in the list

There are different implementations of a list:

- Array list

- Singly linked list

- Doubly linked list

next two videos

IMPLEMENTATIONS

There are different implementations of a list:

- Array list → Use as array to implement a list.

- Singly linked list

- Doubly linked list

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Idea:

- Use an array to store the elements of the list

- Keep track of how many elements we have inserted in the list

To decide:

- How big should the underlying array be when we first create an object of type ArrayList?
  (this is referred to as the *initial capacity* of the list)

ArrayList.java

```
public class ArrayList {
    private Shape[] arr;
    private int size;

    .
    .

}
```

Idea:

- Use an array to store the elements of the list

- Keep track of how many elements we have inserted in the list

To decide:

- How big should the underlying array be when we first create an object of type ArrayList?

→ Java's ArrayList creates an array of length 10.

ArrayList.java

```java
public class ArrayList {
    private Shape[] arr;
    private int size;


    public ArrayList() {
        arr = new Shape[10];
        size = 0;
    }
        :
}
```

ArrayList.java

```
public class ArrayList {
    private Shape[] arr;
    private int size;

    public ArrayList() {
        arr = new Shape[10];
        size = 0;
    }
}
```
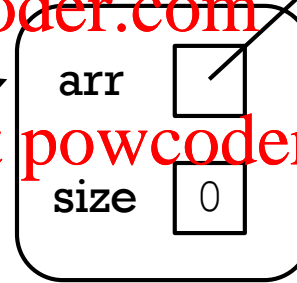
```
ArrayList list = new ArrayList();
```

list

arr

size 0

0 null
1 null
2 null
3 null
4 null
5 null
6 null
7 null
8 null
9 null

ArrayList.java

```java
public class ArrayList {
    private Shape[] arr;
    private int size;

    public ArrayList() {
        arr = new Shape[10];
        size = 0;
    }
}
```
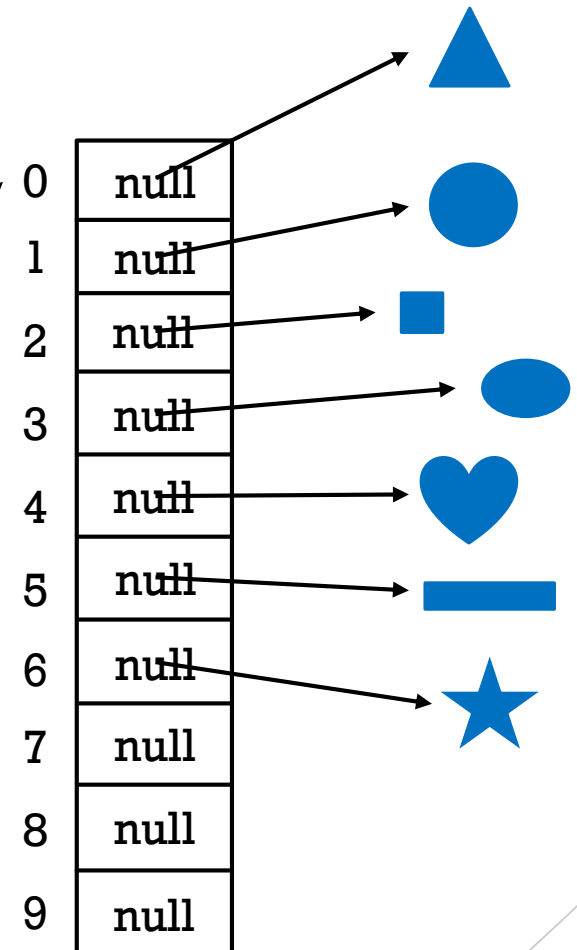
```java
ArrayList list = new ArrayList();
// add 7 elements…
```

list

arr

size 7

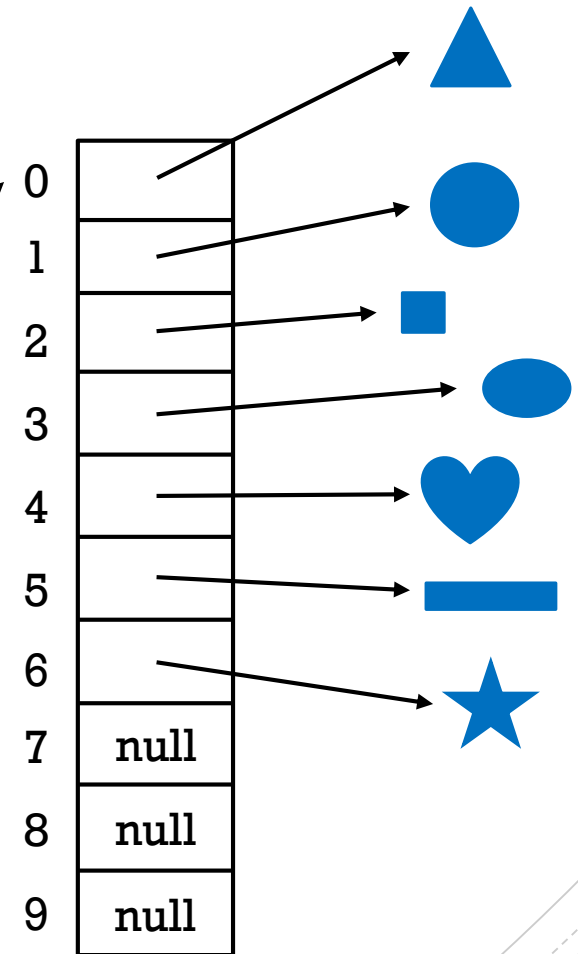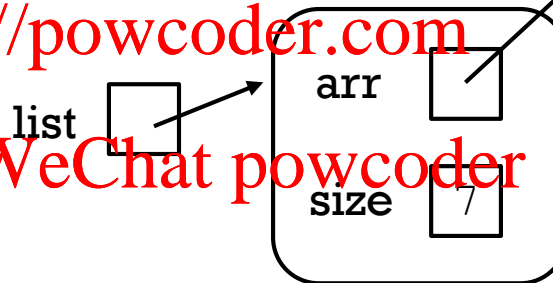| 0 | null |
1	null
2	null
3	null
4	null
5	null
6	null
7	null
8	null
9	null

Returns the element at the specified position in this list.

```
public class ArrayList {
    private Shape[] arr;
    private int size;

    :

    public Shape get(int i) {
        if(                 )
            return arr[i];
        // otherwise?
    }
}
```

list

arr

size 7

0
1
2
3
4
5
6
7 **null**
8 **null**
9 **null**

Returns the element at the specified position in this list.
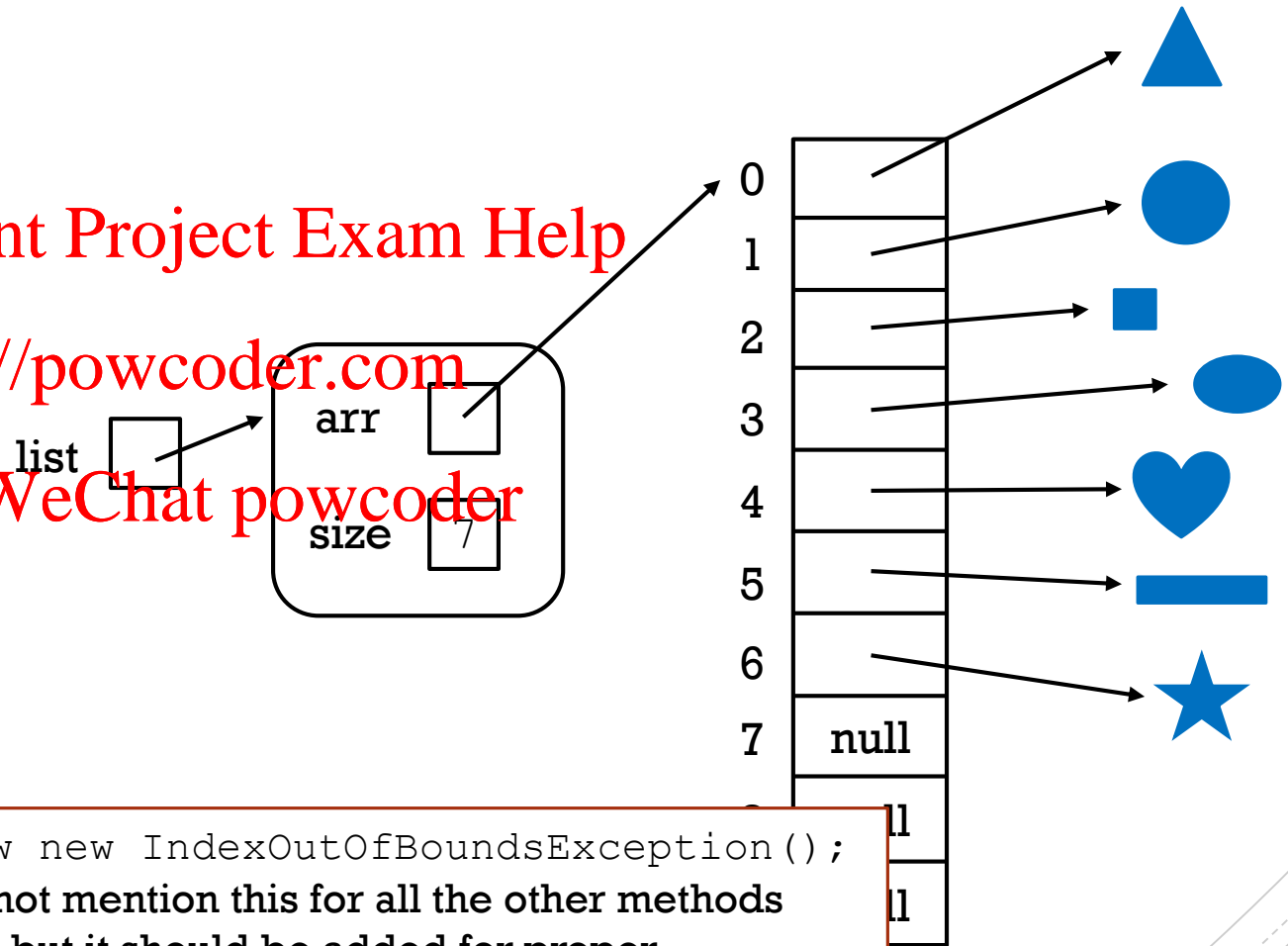
```
public class ArrayList {
    private Shape[] arr;
    private int size;


    :


    public Shape get(int i) {
        if( i>=0 && i<size )
            return arr[i];
        // otherwise?

    }
}
```

list

arr

size   7

0
1
2
3
4
5
6
7   null

null

null

```
throw new IndexOutOfBoundsException();
```
I will not mention this for all the other methods today, but it should be added for proper implementations.

Replaces the element at the specified position in this list with the specified element.

```java
public class ArrayList {
    private Shape[] arr;
    private int size;

    :

    public Shape set(int i, Shape e)
    {
        if(i>=0 && i<size) {
            Shape tmp = arr[i];
            arr[i] = e;
            return tmp;
        }
    }
}
```
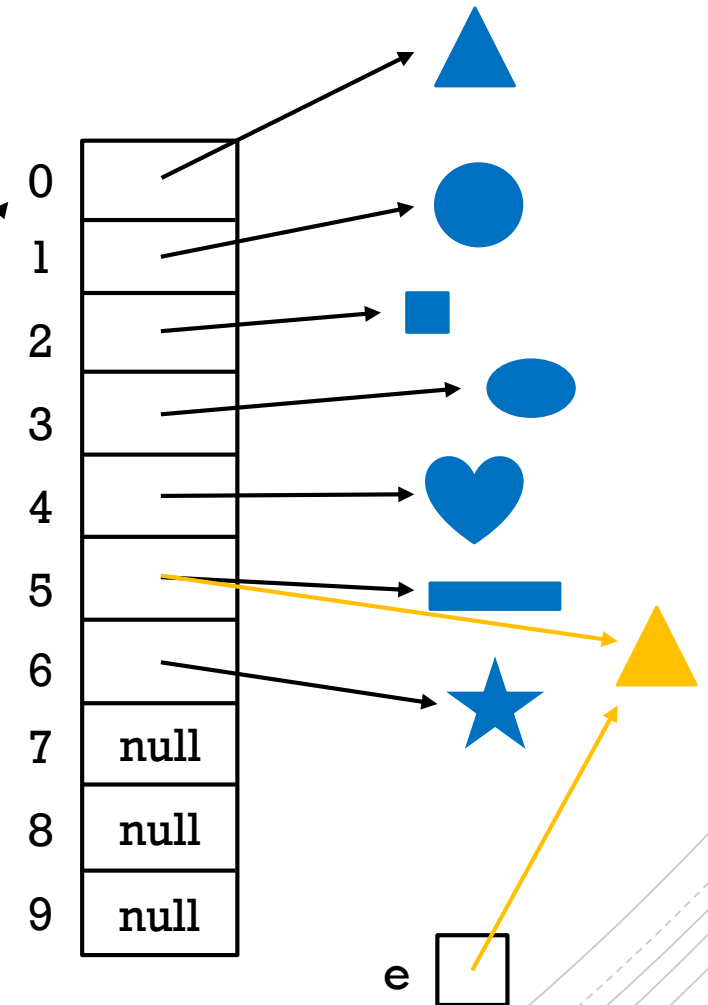
Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

list

arr

size   7

For example:
`list.set(5,e);`

0
1
2
3
4
5
6
7 null
8 null
9 null

e

Appends the specified element to the end of this list.

```
public class ArrayList {
    private Shape[] arr;
    private int size;

    :

    public void add(Shape e)
    {
        arr[    ] = e;
        size = size + 1;
    }
}
```

list

arr

size   7

For example:
`list.add(e);`

| 0 |
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
| 6 |
7	null
8	null
9	null

e

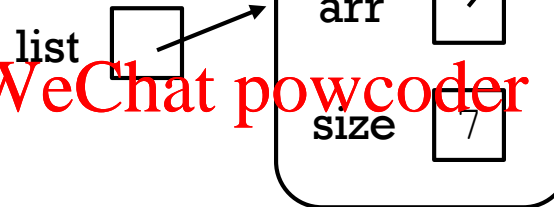**What if the array** `arr` **is full?**

Appends the specified element to the end of this list.

```
public class ArrayList {
    private Shape[] arr;
    private int size;

    :

    public void add(Shape e)
    {
        arr[size] = e;
        size = size + 1;
    }
}
```
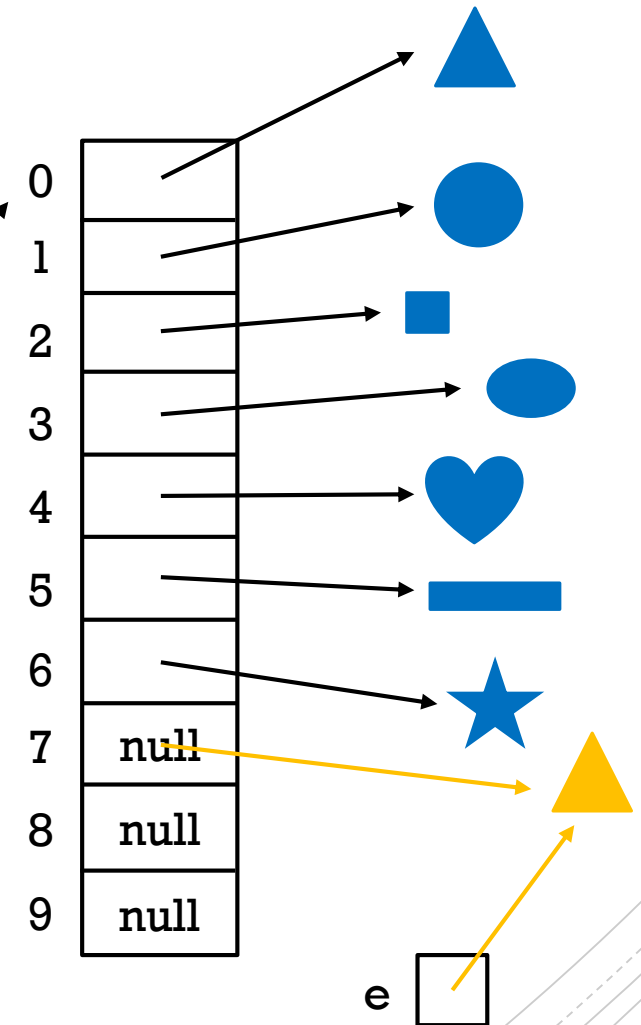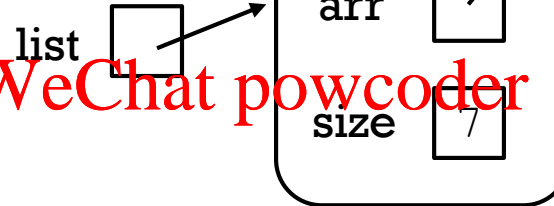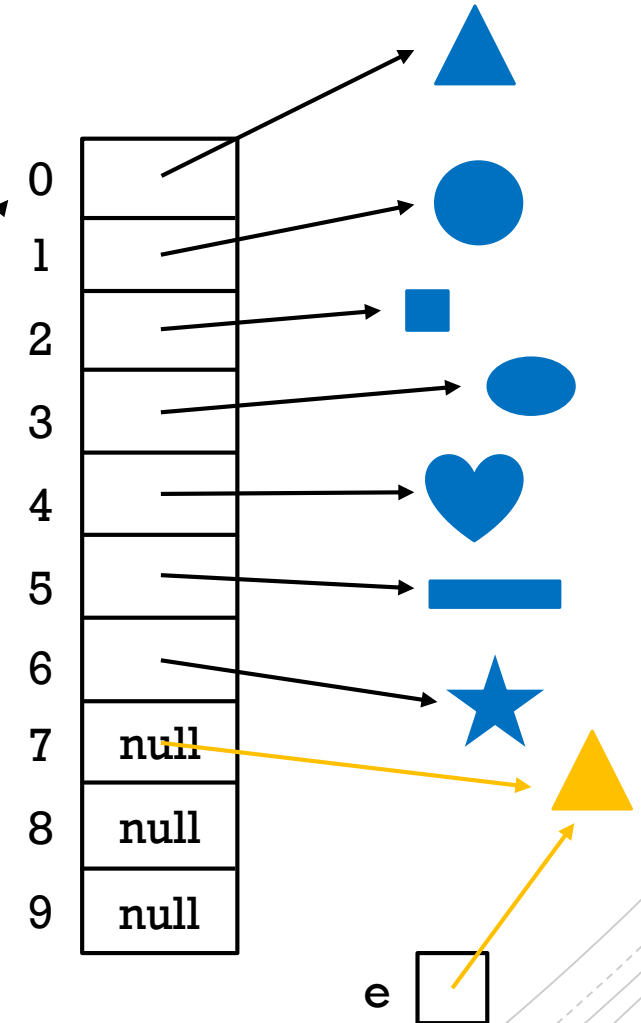
Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

list

arr

size 7

For example:
list.add(e);

0

1

2

3

4

5

6

7 null

8 null

9 null

e

What if the **array** `arr` **is full?**

Appends the specified element to the end of this list.

What if the array arr is already full?

```java
public void add(Shape e) {
    if (                              )
        resize();
    arr[size] = e;
    size = size + 1;
}

private void resize() {
    Shape[] bigger = new Shape[arr.length*2]; // example
    for(int i=0; i < size; i++) {
        bigger[i] = arr[i];
    }
    arr = bigger;
}
```

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Appends the specified element to the end of this list.

What if the array arr is already full?

```
public void add(Shape e) {
    if (arr.length == size)
        resize();
    arr[size] = e;
    size = size + 1;
}


private void resize() {
    Shape[] bigger = new Shape[arr.length*2]; // example
    for(int i=0; i < size; i++) {
        bigger[i] = arr[i];
    }
    arr = bigger;

}
```

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

add(e)   // inserts element e at end of list

add(i ,e)

// Inserts element e into the i-th position

remove(i)   // Removes the i-th element from list
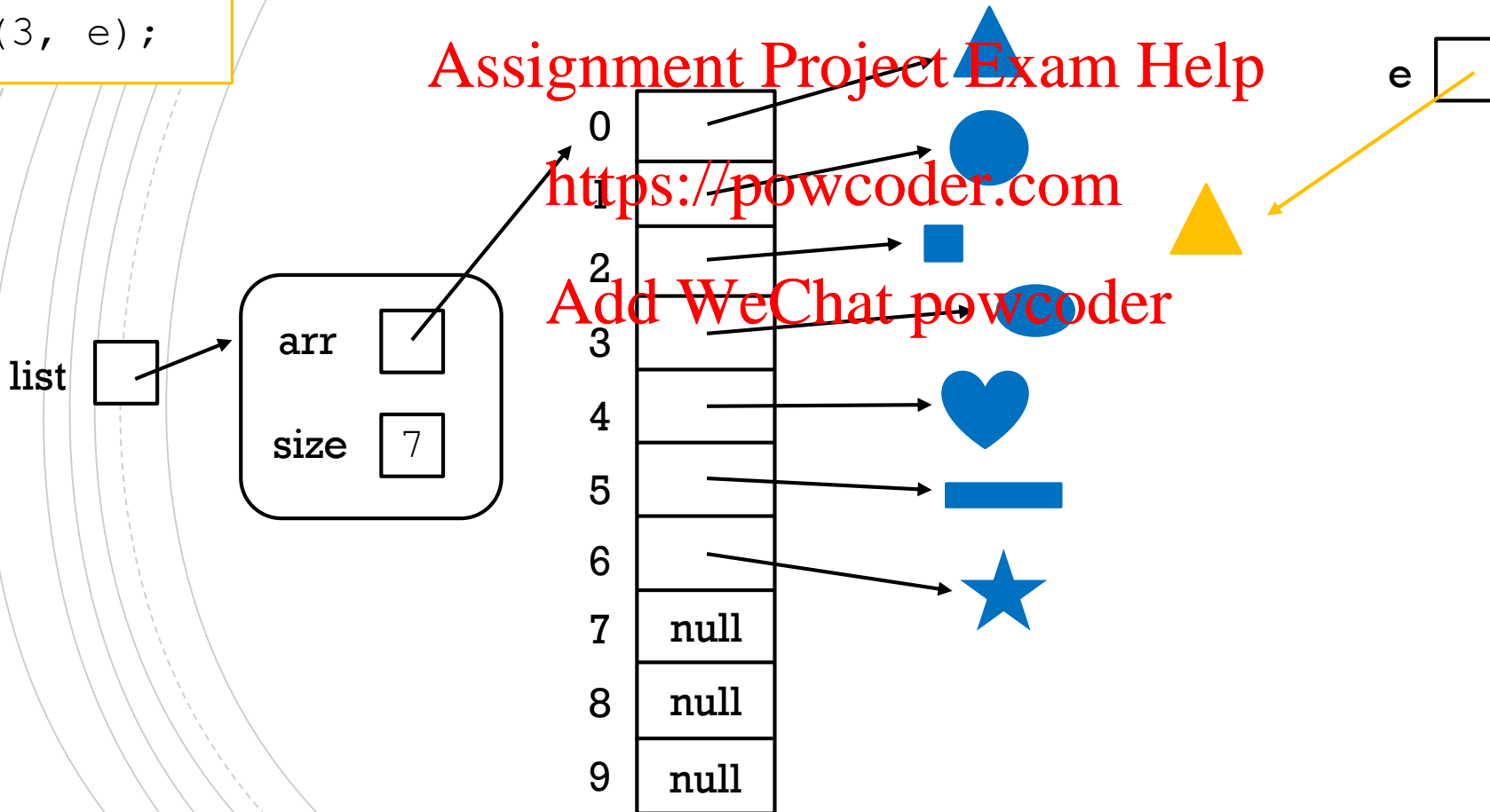
remove(e)   // Removes first occurrence of element e

                  //   from the list (if it is there)

Inserts the specified element at the specified position in the list.

For example:
```
list.add(3, e);
```

e

list

arr

size 7

0
1
2
3
4
5
6
7 null
8 null
9 null
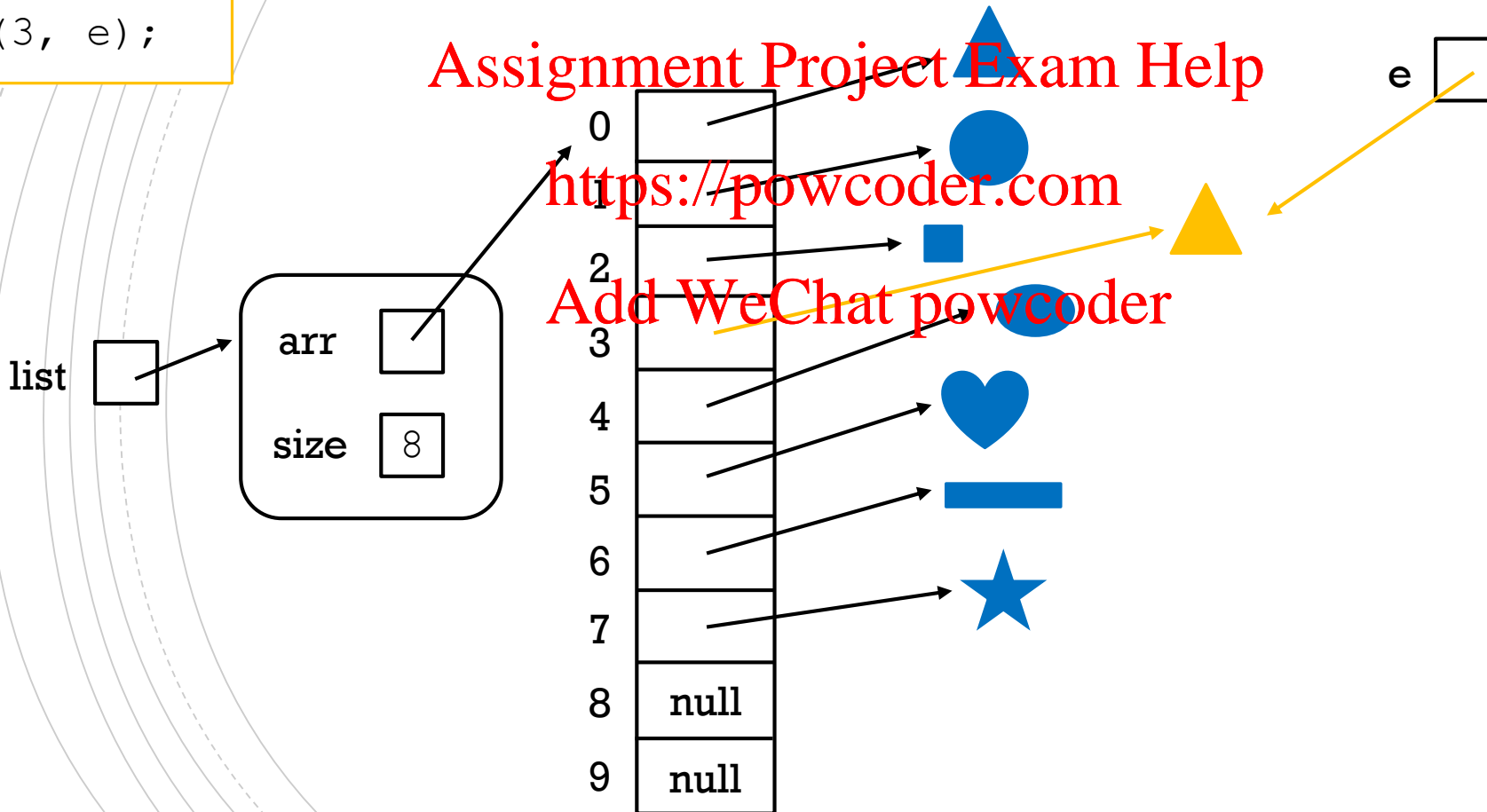
Inserts the specified element at the specified position in the list.

IDEA: make room by shifting the elements, then add e

For example:

```
list.add(3, e);
```

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

e

list

arr

size 8

0

1

2

3

4

5

6

7

8 null

9 null

Inserts the specified element at the specified position in the list.

```
public void add(int i, Shape e) {



    // Throw exception if i is out of bounds



    // Resize if not enough space



    // Shift elements down



    // Add the new element



}
```

Suppose we initialize an array list with an empty array of length 1.
We then add an element.

arraylist of
size 0
(length 1)

arraylist of
size 1
(length 1)

What do we do to add
a second element?
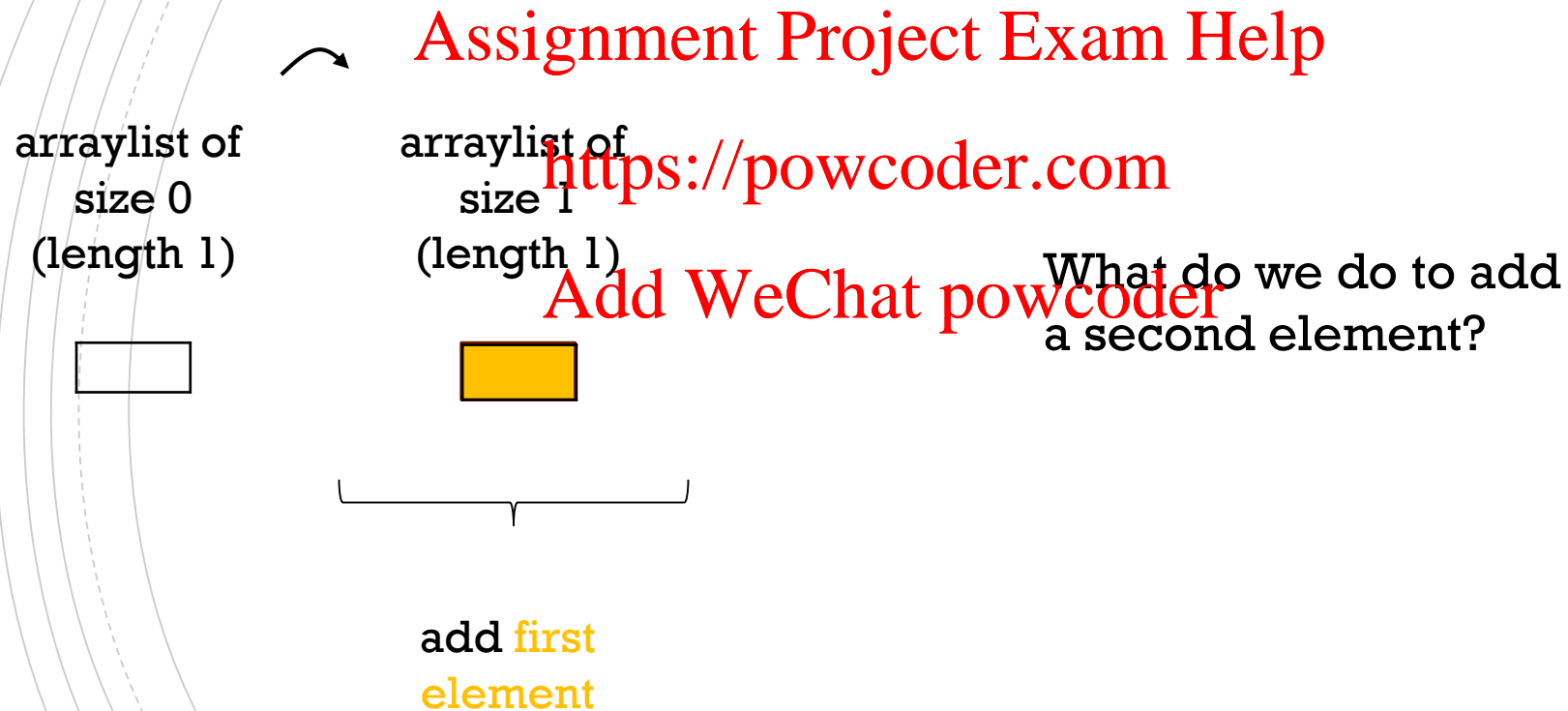
add first
element

Suppose each time we add to a full array list, we double the length of the array.

Assignment Project Exam Help

https://powcoder.com

arraylist of size 1 (length 1)

arraylist of size 1 (length 2)

arraylist of size 2 (length 2)
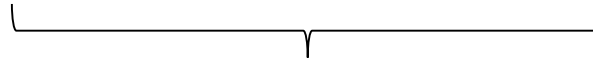
Add WeChat powcoder

add second element

ADDING N ELEMENTS TO AN ARRAY LIST

Assignment Project Exam Help

arraylist of size 2 (length 2)

arraylist of size 2 (length 4)

arraylist of size 3 (length 4)

https://powcoder.com

Add WeChat powcoder

add third element

arraylist of size 3 (length 4)

arraylist of size 4 (length 4)

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

add *fourth element*

ADDING N ELEMENTS TO AN ARRAY LIST

arraylist of
size 4
(length 4)

arraylist of
size 4
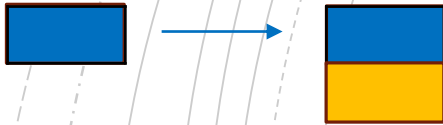(length 8)

arraylist of
size 5
(length 8)

add fifth element

ADDING N ELEMENTS TO AN ARRAY LIST

Double length and
copy one element

Double length and
copy two elements

Double length and
copy four elements

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

add two elements

add 3-4 elements

add 5-8 elements

Q: How many times k do we need to double the length of the array so that it is of length N ?

A:

Q: How many copy operations are required to add N elements to an empty arraylist ?

A:

Q: How many times k do we need to double the length of the array so that it is of length N ?

A: $2 \cdot 2 \cdots 2 = 2^k = N \rightarrow k = \log_2 N$

Q: How many copy operations are required to add N elements to an empty arraylist ?

A:

Q: How many times k do we need to double the length of the array so that it is of length N ?

A: $\quad 2 \cdot 2 \cdots 2 = 2^k = N \;\rightarrow\; k = \log_2 N$

Q: How many copy operations are required to add N elements to an empty arraylist ?

A: $1 + 2 + 4 + \ldots + 2^{k-1}$

- *Geometric series*

$$\sum_{i=0}^{n} x^i = 1 + x + x^2 + \cdots + x^n = \frac{1 - x^{n+1}}{1 - x}$$

- *Arithmetic series*

$$\sum_{i=1}^{n} i = 1 + 2 + \cdots + n = \frac{1}{2} n(n + 1)$$

Q: How many times k do we need to double the length of the array so that it is of length N ?

A: $2 \cdot 2 \cdots 2 = 2^k = N \rightarrow k = \log_2 N$

Q: How many copy operations are required to add N elements to an empty arraylist ?

A: $1 + 2 + 4 + \ldots + 2^{k-1} = \sum_{i=0}^{k-1} 2^i = \frac{1-2^k}{1-2} = 2^k - 1 = N - 1$

https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html

Assignment Project Exam Help

- It uses an array as the underlying data structure

    https://powcoder.com

- It grows the array (by 50%, Add 100%() when the array is full and a new element is added.
    Add WeChat powcoder

- As a client, we don't have access to the fields directly. We need to use the methods provided (get(), set(), …) to manipulate the list.

- `ArrayList` **is a generic class with a type parameter.**

- **When you create an object of type** `ArrayList` **you specify the type of the elements stored by the list by appending to** `ArrayList` **a class name enclosed in angle brackets.**

- **Example:**

> If we write `int` instead we get a compile-time error!

```
// creates an arraylist of integers with initial capacity 10
ArrayList<Integer> words = new ArrayList<Integer>();

// creates an arraylist of shapes with initial capacity 23
ArrayList<Shape> myShapes = new ArrayList<Shape>(23);
```

WRAPPER CLASSES

- `Integer`, `Double`, and `Character` **wrap a value of the primitive type** `int`, `double`, **and** `char` **(respectively) in an object. Thus, they turn primitive types into reference types.**

- **The conversion between the primitive types and their wrappers is done automatically. For example, the following would not cause a compile-time error:**

```
Integer x = 5;
```

- **Note that these classes have static methods/attributes that you might have already used. For example:** `Integer.MAX_VALUE`, `Double.parseDouble()`

AUTOBOXING AND UNBOXING

- *Autoboxing* is the automatic conversion that the Java compiler makes between the primitive types and their corresponding object wrapper classes. For example converting an int to an Integer.

```
Integer x = 5;
```

- If the conversion goes the other way, this is called *unboxing*.

```
Integer x = new Integer(5);
int y = x;
```

- https://docs.oracle.com/javase/tutorial/java/data/autoboxing.html

- Note that `Integer`, `Double`, **and** `Character` **are immutable** reference types (like `String`).

- As with `String`, **you can appear to updates values**, but you are never changing the actual Object. A new Object gets created each time we "change" a value.

It is much simpler (in terms of code re-use) to have `ArrayList` require the input to be an Object (a reference type), instead of using primitive types.

For example, all reference types can be compared using `.equals()`, while we have to use `==` for primitive types.

```
int[] numbers = {1, 2, 3, 4, 5};

for(int element: numbers) {

    System.out.println(element);

}
```

The foreach loop (also called enhanced for loop) can make your code more readable and can be convenient to use. It is not helpful when you need to refer to the index of an element.

Coming Soon

Assignment Project Exam Help

In the next videos:

https://powcoder.com

▪ Linked lists

Add WeChat powcoder