Last Time

- our SmartArray → ArrayList
- ArrayList
  - useful
  - no need to memorize. add to cheat sheet
- Generics. We write:
  ```java
  ArrayList<type> name = new ArrayList<type>();
  e.g.,
  ArrayList<String> name = new ArrayList<String>();
  ```

Suppose that we have A, which is a two-dimensional array of int. Which of the following gives us the item in A which is stored in the last row, first column?

- A[0][A[0].length-1]
- A[length][0]
- A[A.length-1][0]
- A[0].length-1
- Impossible to tell with the information given.
Suppose that we have $A$, which is a two-dimensional array of `int`. Which of the following gives us the item in $A$ which is stored in the last row, first column?

- $A[0][A[0].length-1]$
- $A[length][0]$
- $**** A[A.length-1][0] ****$
- $A[0].length-1$
- Impossible to tell with the information given.

```java
public class WhatsPrinted {
    public static void func(int A[], int x) {
        for (int i=0; i<A.length; i++)
            A[i]+=x;
    }
    public static void main(String args[]) {
        int []A = {10,20,30,40};
        func(A, 2);
        System.out.println(A[A.length-1]);
    }
}
```

**Answer:** 77
```java
public class WhatsPrinted {
    public static void mystery(int y, int z) {
        if (y+3 > z) {
            y++;
        } else {
            y--;
        }
    }
    public static void main(String args[]) {
        int x=10;
        int y=20;
        int z=30;
        mystery(x, y);
        mystery(y, z);
        System.out.println(y);
    }
    public static void func(int A[], int x) {
        int B[] = new int[A.length];
        for (int i=0; i<A.length; i++) {
            B[i]=A[i]*x;
        }
        A=B;
    }
    public static void main(String args[]) {
        int arr[]={31,20,18,5};
        func(arr, 2);
        System.out.println(arr[0]);
    }
}
```

```java
public class WhatsPrinted {
    public static void func(int A[], int x) {
        int B[] = new int[A.length];
        for (int i=0; i<A.length; i++) {
            B[i]=A[i]*x;
        }
        A=B;
    }
    public static void main(String args[]) {
        int arr[]={31,20,18,5};
        func(arr, 2);
        System.out.println(arr[0]);
    }
}
```

**Answer:** 31

```java
class Stuff {
    int x;
    /* constructor */
    public Stuff(int newX) {
        x=newX;
    }
}
```

```java
public class WhatsPrinted {
    public static void func(Stuff s) {
        s.x*=2;
    }
    public static void main(String args[]) {
        Stuff s1 = new Stuff(10);
        Stuff s2 = new Stuff(4);
        func(s2);
        s2.x+=2;
        if (s1 == s2) {
            System.out.println("same");
        } else {
            System.out.println("different");
        }
    }
}
```

**Answer:** different

Which analogy is most accurate?

- cookie cutter is to cookie as object is to class
- cookie is to cookie cutter as object is to class
- cookie cutter is to cookie as blueprint is to object
- cookie is to cookie cutter as blueprint-class is to class
- cookie is to cookie cutter as base class is to object

**Answer:** different
Which analogy is most accurate?

► cookie cutter is to cookie as object is to class
► *** cookie is to cookie cutter as object is to class ***
► cookie cutter is to cookie as blueprint is to object
► cookie is to cookie cutter as blueprint-class is to class
► cookie is to cookie cutter as base class is to object

Answer: 12
public class SweDihShh {
    public static void main(String args[]) {
        String s1="swe";
        String s2="dih";
        String s3="shh";

        for (int i=0; i<5; i++) {
            if (i%2==0) {
                s2+=s3;
            } else {
                s1+=s2;
            }
        }
        System.out.println(s1);
    }
}

Answer: swedihshhdihshhshh

public static int[]
    scale(int A[], int f) {
        int []newA = new int[A.length];
        for (int i=0; i<A.length; i++) {
        }
        return newA;
    }

public static void
    main(String args[]) {
        int []A = {15,30,45,60,75,90};
        int []B = {5,10,15,20,25,30};
        C=A;
        D=scale(B, 3);
        if (A==B) {
            System.out.print("s");
        } else {
            System.out.print("d");
        }
        if (A==C) {
            System.out.print("s");
        } else {
            System.out.print("d");
        }
        if (A==D) {
            System.out.print("s");
        } else {
            System.out.print("d");
        }
    }
}

Answer: dsd

Which of the following methods returns the location of the largest element in A? You may assume that A contains at least one element?
Which of the following methods returns the location of the largest element in A? You may assume that A contains at least one element?

**Answer:**

```java
class IndexOfLargest {
    public static int indexOfLargest(int[] A) {
        int L = 0;
        for (int i = 1; i < A.length; i++) {
            if (A[i] > A[L]) {
                L = i;
            }
        }
        return L;
    }
}
```

Write a method same which is passed two arrays of int. The method returns true if the contents of both arrays is identical.

```java
public static boolean same(int A[], int B[]) {
    if (A.length != B.length) {
        return false;
    }
    for (int i = 0; i < A.length; i++) {
        if (A[i] != B[i]) {
            return false;
        }
    }
    return true;
}
```

Write a method called elementOf which is passed a String s and a char c. The method returns true if c is found in s or false otherwise.

```java
public static boolean elementOf(String s, char c) {
    for (int i = 0; i < s.length(); i++) {
        if (s.charAt(i) == c) {
            return true;
        }
    }
    return false;
}
```
Write a method called `elementOf` which is passed a `String s` and a `char c`. The method returns `true` if `c` is found in `s` or `false` otherwise.

```java
public static boolean elementOf(String s, char c) {
    for (int i=0; i<s.length; i++) {
        if (s.charAt(i)==c)
            return true;
    }
    return false;
}
```

Write a method called `rmchars`, which is passed two `String`s, `str`, and `chars`. The method returns a new `String`, which is exactly the same as `str`, but with all of the letters in `chars` removed. For example, if we call `rmchars` where `str` is "You’re going to need a bigger boat." and `chars` is "Roy", the method returns "Yu’re ging t need a bigger bat."

```java
public static String rmchars(String str, String chars) {
    String r="";
    for (int i=0; i<str.length(); i++) {
        char c = s.charAt(i);
        if (!elementOf(chars, c)) {
            r+=c;
        }
    }
    return r;
}
```

Write the single line of code that calls your method from part b in order to remove the letters 'a' and 'r' from the String "Shark lasers" saving the result in a String variable named `s`. 

Write a method called `elementOf` which is passed a `String s` and a `char c`. The method returns `true` if `c` is found in `s` or `false` otherwise.
Write the single line of code that calls your method from part b in order to remove the letters 'a' and 'r' from the String "Shark lasers" saving the result in a String variable named s.

String s = rmchars("Shark lasers", "ar");