AP CS Unit 5: Arrays
Exercises

1. What is displayed? 3	int [] a = new int[3]; System.out.println(a.length);
2. What is displayed? 23 12 156	int [] sting = { 34, 23, 67, 89, 12 }; System.out.println(sting[1]); System.out.println(sting[sting.length - 1]); System.out.println(sting[2] + sting[3]); 67 + 89

3. What is displayed?
-9
0.0
5.2
14.5
	double [] ray = new double[4];
	ray[2] = 5.2;
	ray[0] = -9;
	ray[3] = 14.5;
	ray[1] = 0.0;
	
	for (int n = 0; n < ray.length; n++)
	System.out.println(ray[n]);
	ray [ ] → -9, 0.0, 5.2, 14.5

4. What is displayed?
0
3
6
9
	int [] arr = new int[4];
	for (int k = 0; k < arr.length; k++)
	arr[k] = 3*k;
	
	for (int k = 0; k < arr.length; k++)
	System.out.println(arr[k]);
	arr [ ] → 0, 3, 6, 9

5. What is displayed?
3
	int [] arr = { 5, 2, 3, 5, 2, 4, 5 };
	int x = 0;
	for (int n = 0; n < arr.length; n++) {
	if (arr[n] == 5)
	x++;
	}
	System.out.println(x);
	x: 0 1 2 3 ; n0 → 5, n3 → 5, n6 → 5

6. What is displayed?
1
 	
note: break only exits loops!
	int [] arr = { 5, 2, 2, 2, 2, 4, 5 };
	int x = -1;
	for (int n = 0; n < arr.length; n++) {
	if (arr[n] == 2){
	x = n;
	break;
	}
	}
	System.out.println(x);
	x: -1 → 1 ; 0 → 5, 1 → 2, 2 → 2, 2 → 2, 2 → 2, 4 → 4, 5 → 5

arr → | 14 | 13 | 12 | 11 | 10 |

k = 4 to 0

7. What is displayed?
 	
14
13
12
11
10
	int [] arr = new int[5];
	for (int k = arr.length - 1; k >= 0; k--)
	arr[4 - k] = k + 10;
	
	for (int k = 0; k < arr.length; k++)
	System.out.println(arr[k]);

8. What is displayed?
 	
8
 	
basic summation
	int x = 0;
	int [] arr = { 5, -5, 7, 1 };
	for (int n = 0; n < arr.length; n++)
	x = x + arr[n];
	
	System.out.println(x);

9. This compiles. What is displayed? If there is a run-time error then indicate what the problem is.
 	
don't see the error
 	
0
4
	public class Runner{
	public static void main(String [] args){
	int [] a = { 7, 3, 5, 1 };
	0 System.out.println(met(a));
	int [] b = { 4, 8, 6, 6, 9 };
	4 System.out.println(met(b));
	}
	
	private static int met(int [] a){
	for (int k = 0; k < a.length; k++){
	if (a[k] % 2 != 0)
	return k;
	}
	return -1;
	}
	}

10. Select the TRUE statement.
a) This code will not compile. The error will be: *ArrayIndexOutOfBoundsException*
b) This code will compile but if you try to run it, the following run-time exception will occur:
ArrayIndexOutOfBoundsException
c) This code will compile and run. It will display a zero.
	public class Runner{
	public static void main(String [] args){
	int [] c = { 7, 6, 5 };
	met(c);
	}
	
	private static void met(int[] a){
	System.out.println(a[4444]);
	}
	}

11. Select the TRUE statement.
a) This code will not compile. The error will be: *local variable c may not have been initialized*
b) This code will compile but if you try to run it, the following run-time exception will occur:
NullPointerException
c) This code will compile and run. It will display a zero.
	public class Runner{
	public static void main(String [] args){
	int [] c;
	met(c);
	}
	
	private static void met(int[] a){
	System.out.println(a.length);
	}
	}

12. What is displayed?

3
1.0
300
149.5
0

```
public class Runner{
    public static void main( String [] args ){
        double [] c = met( 3 );
        System.out.println( c.length );
        System.out.println( c[ c.length-1 ] );
        double [] d = met( 300 );
        System.out.println( d.length );
        System.out.println( d[ d.length-1 ] );
        double [] e = met( 0 );
        System.out.println( e.length );
    }

    private static double [] met( int n ){
        double [] d = new double[n];
        for ( int k = 0; k < n; k++ ){
            d[k] = k / 2.0;
        }
        return d;
    }
}
```

13. What is displayed when this client code is executed?

```
Sack sam = new Sack( 3 );
sam.add( 8 );
sam.add( 3 );
System.out.println( sam.toString() );
```

depending on IDE either prints null or weird stuff

sam → s → 8 3 0, count: 0 1 2

no 3's go in bc count == s.length

→

→ 8 3 0

```
public class Sack{
    private int[] s;        // holds null ptr at first
    private int count;

    public Sack( int n ){
        System.out.println( s );
        s = new int[ n ]; <- creates array of 0's
        count = 0;
    }

    public void add( int x ){
        if ( count < s.length ){
            s[count] = x;
            count++;
        }
    }

    public String toString(){
        String y = "";
        for ( int k = 0; k < s.length; k++ )
            y += s[k] + " ";

        return y;
    }
}
```

14. Select the TRUE statement.

a) This code will not compile. The error will be: *local variable c may not have been initialized*

b) This code will compile but if you try to run it, the following run-time exception will occur:
NullPointerException

c) This code will compile but if you try to run it, the following run-time exception will occur:
variable c has not been initialized

d) This code will compile and run. It will display a zero.

```
public class Runner{
    public static void main( String [] args ){
        int [] c = null;
        met( c );
    }

    private static void met( int[] a ){
        System.out.println( a.length );
    }
}
```

15. The code to the right contains a common mistake. It compiles and the constructor runs. However, if you try to run the get method you will always get the following error:

NullPointerException

What is the problem and how do you fix it?

you remove int[] inside public Satchel

```
public class Satchel{
    private int [] stuff;

    public Satchel( int x ){
        int[] stuff = new int[ x ];
        for ( int k = 0; k < x; k++ )
            stuff[k] = k + 1;
    }

    public int get( int index ){
        return stuff[ index ];
    }
}
```

int[] stuff local

16. This compiles and runs. What is displayed when the mm6 method is called? This is a little tricky.

4

```
public void mm6() {
        int [] b = { 4, 5 };
        mxx( b );
        System.out.println( b[0] );
}

public void mxx( int [] z ){
        int [] a = {7, 8};
        z = a;
}
```

17. This compiles and runs. What is displayed when the mx4 method is called?

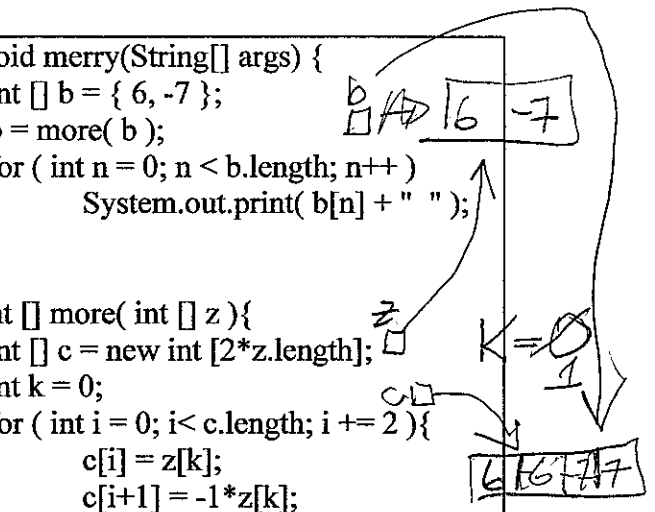
34

```
public void mx4() {
        int [] a = {9, 8, 7 };
        mx5( a );
        System.out.println( a[0] );
}

public void mx5( int [] z ){
        z[0] = 34;
}
```

18. This compiles and runs. What is displayed when the merry method is called?

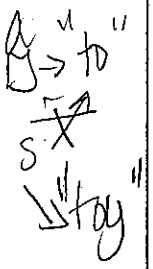
6, -6, -7, 7

```
public void merry(String[] args) {
        int [] b = { 6, -7 };
        b = more( b );
        for ( int n = 0; n < b.length; n++ )
                System.out.print( b[n] + "  " );
}

public int [] more( int [] z ){
        int [] c = new int [2*z.length];
        int k = 0;
        for ( int i = 0; i< c.length; i += 2 ){
                c[i] = z[k];
                c[i+1] = -1*z[k];
                k++;
        }
        return c;
}
```



19. What is displayed when this program is run? This is tricky too.

"to"




```
public class General {
        public static void main(String[] args) {
                String a = "to";
                m9( a );
                System.out.println( a );
        }

        public static void m9( String s ){
                s = s + "y";
        }
}
```

20. Instantiate an array of 20 RandNum objects.

Calculate the sum of their numbers.

```
RandNum[] rands = new RandNum[20];
int sum=0;
for(int i=0; i<rands.length; i++){
        rands[i] = new RandNum();
        sum += rands[i].getNum();
}
```

```
public class RandNum {

    private int num;

    public RandNum() {
        num = (int)(10*Math.random());
    }

    public int getNum(){
        return num;
    }
}
```

21. Instantiate an array of 200 good Dog objects and make each dog bark.

```
Dog[] dogs = new Dog[200];
for (int i=0; i< dogs.length; i++) {
    dogs[i] = new Dog(true);
    dogs[i].bark();
}
```

```
public class Dog {
    // private instance variables

    public Dog( boolean goodDog ) {
        // implementation code
    }

    public void bark(){
        // implementation code
    }

    // possibly other methods
}
```

Enhanced for loop section"

22. What is displayed?

5.5
6
1.3
3.1

```
double [] xray = { 5.5, 6, 1.3, 3.1 };
for ( double ele : xray )
        System.out.println( ele );
```

23. What is displayed?

3 0 -2 1 3 4

n = 2 3 4 5

0	1	2	3	4	5
3	0	-2	1	3	4

n to 5

```
int [] ray = { 8, 2, 5, 5, 3, 4 };
for ( int n = 2; n < ray.length; n++ )
        ray[ n – 2 ] = ray[n] – ray[ n – 1 ];

for ( int n = 0; n < ray.length; n++ )
        System.out.print( ray[n] + "\t ");
```

24. What is displayed?

false
false

```
boolean [] cosmic = { false, true, true, false };
for ( boolean mem : cosmic )
        if ( !mem )
                System.out.println( mem );
```

25. The Ox class compiles. Does the client code compile and run? If yes, what is displayed? If no, what is the problem?

6
2
3

```
// client code
Ox [] a = new Ox[3];
a[2] = new Ox( 5 );
a[0] = new Ox( 6 );
a[1] = new Ox( 2 );

for ( Ox babe : a )
    System.out.println( babe.get());
```

a → | 0 | 1 | 2 |

```
public class Ox{
    private int x;

    public Ox( int k ){
        x = k;
    }

    public int get(){
        return x;
    }
}
```

26. Complete the code so that it accurately counts the number of even numbers in the array.

```
int [] ray = { 8, 2, 5, 5, 3, 4 };
int num = 0;

for ( int ar : ray ) {
    if(ar % 2 == 0)
        num++;
}
System.out.println( "There are " + num + " even numbers." );
```

27. What is displayed?

11

```
String [] s = {"it", "can", "be", "good" };
int total = 0;
for ( String wd : s )
        total += wd.length();

System.out.println( total );
```

2 3 2 4

28. What is displayed?

3
4

```
int [][] ray = new int[3][4];
System.out.println( ray.length );
System.out.println( ray[0].length );
```

0 1 2 3
0
1
2

29. What is displayed?

0 0 0 0
0 1 2 3
0 2 4 6

```
int [][] ray = new int[3][4];
for ( int row = 0; row < 3; row++ )
        for ( int col = 0; col < 4; col++ )
                ray[ row ][ col ] = row*col;

for ( int row = 0; row < 3; row++ ) {
        for ( int col = 0; col < 4; col++ )
                System.out.print(ray[ row ][ col ] + " ");
        System.out.println( );
}
```

	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	4	6

30. What is displayed? This is a little tricky.

table

A	C	E	G
B	D	F	H

careful rows changes in inner loop

A C E G B D F H

```
String [] ltrs = { "A", "B", "C", "D", "E", "F", "G", "H" };
String [][] table = new String[2][4];
int n = 0;

for ( int col = 0; col < table[0].length; col++ ) {
        for ( int row = 0; row < table.length; row++ ){
                table[ row ][ col ] = ltrs[ n ];
                n++;
        }
}

for ( int row = 0; row < table.length; row++ ) {
        for ( int col = 0; col <  table[0].length; col++ )
                System.out.print( table[ row ][ col ] + " ");
        System.out.println( );
}
```

col: 0
row: 0

31. What is displayed?

3
2
8

```
int [][] a = { {6, 1, 1}, {8, 2, 5} };
System.out.println( a[0].length );
System.out.println( a.length );
System.out.println( a[1][0] );
```

a →

6	1	1
8	2	5

if we add: c[0][0] = new Cat(); c[1][0] = new Cat(); then it works

c → [null | null]

32. This code compiles but throws the following runtime exception:
NullPointerException
Why? you did not instantiate any Cat objects. Cat[a][0] = new Cat();
	Cat [][] c = new Cat[2][1];
	for (Cat [] d : c) {
	for (Cat e : d)
	System.out.print(e.toString() + " ");
	System.out.println();
	}
	d cat object c[]
	e cat object

33. Complete the program so that the minimum value of each row in the 2D array is displayed within the loop. You must use the findMin method.	public class Runner{
	public static void main(String [] args){
	int [][] a = new int[5][5];
	for (int r = 0; r < 5; r++){
	for (int c = 0; c < 5; c++){
	a[r][c] = (int)(101*Math.random());
	}
	}
	for (int r = 0; r < 5; r++){
	int myMin = findMin(a[r]);
	System.out.println(myMin);
	}
	}
	
	private static int findMin(int [] a){
	int min = a[0];
	for (int k = 1; k < a.length; k++){
	if (a[k] < min)
	min = a[k];
	}
	return min;
	}
	}

34. Complete the method below. It returns an array of ints that is a copy of the column in table specified by col. For example:

	0	1	2
0	6	3	5
1	2	9	4

If this is the table and col =0 then the method returns an array that contains 6 and 2.

Assume that col contains a valid value.

```
public int [] getColumn( int [][] table, int col ) {
    int[] newArr = new newArr[table.length];
    for (int i = 0; i < table.length; i++) {
        newArr[i] = table[i][col];
    }
    return newArr;
```