AP CS Unit 5.1: Static Variables and Methods Exercises

Static Methods and Static Variables.

Problems 1 to 3 assume that the mentioned code is in the main method of the PiggyWiggy class.

1. Will this code compile? _____

```
System.out.println( Pig.talk() );
```

2. Will this code compile? _____

```
int w = Pig.getWeight();
System.out.println( w );
```

3. This code contains a syntax error.

```
System.out.println( Pig.MAX_WT() );
```

What is it? _____

4. If we added the following method to the Pig class, would it compile? _____

```
public static String squeal() {
    return "I weigh " + wt + " pounds.";
}
```

```
public class PiggyWiggy {
    public static void main(String [] args)
{
        // code referred to problems 1 to 3
    }
}
```

```
public class Pig {
    private int wt;
    public static final int MAX_WT=900;

    public Pig( int w ) {
        if ( w > MAX_WT )
            wt = MAX_WT;
        else
            wt = w;
    }

    public int getWeight() {
        return wt;
    }

    public static String talk() {
        return "oink";
    }
}
```

5. This code does not compile. Why?

6. Fix the problem so that we can call the gogogo method. What will be displayed?

```
public class Question {
        public static void main( String [] args ){
                int n = gogogo( 8.3 );
                System.out.println( n );
        }

        public int gogogo( double x ){
                return (int) ( x + 0.5 );
        }
}
```

7. A static variable can never be accessed in an instance method.	True	False
8. An instance variable can never be accessed in a class method.	True	False

9. The Fish class compiles. If we changed m5 from an instance method to a class method (by adding the keyword *static*), would it still compile?

```
public class Fish{
        public int m5( int y ){
                int x = y + 7;
                return x;
        }
}
```

10. The Tuna class compiles. If we changed m6 from a class method to an instance method (by removing the keyword *static*), would it still compile?

```
public class Tuna{
        public static int m6( int y ){
                int x = y + 7;
                return x;
        }
}
```

11. What is displayed?

```
public class Runner{
    public static void main( String [] args ) {
        Fowl f1 = new Fowl();
        Fowl f2 = new Fowl();
        f2.m2();
        f1.m2();
        f2.m2();
        f1.m3();   _____
        f2.m3();   _____
    }
}
```

```
public class Fowl{
    private static int a = 3;
    private int b = 10;

    public void m2(){
        a++;
        b++;
    }

    public void m3(){
        System.out.println(a + "," + b);
    }
}
```

12. After the two lines to the right are executed,

how many instance variables exist? _____

How many class variables exist? _____

```
Fowl f1 = new Fowl();
Fowl f2 = new Fowl();
```

13. Select the TRUE statement(s). (Assume the code compiles.)

a) x and y must be instance variables.
b) x and y must be class variables.
c) x and y might both be instance, class variables, or one of each.
d) x and y are local variables.

```
public int methodX() {
        return x + y;
}
```

14. Select the TRUE statement.

a) The method does not compile. It would compile if the keyword static were removed.
b) The method compiles. It would not compile if the keyword static were removed.
c) The method compiles and it would still compile if the keyword static were removed.
d) The method does not compile and it would still not compile even if the keyword static were removed.

```
public class Pig {
    private int x = 5;
    private static int y = 7;

    public static void m77() {
        x++;
        y--;
    }
}
```

15. Complete the close method so that it returns true if the difference between n1 and n2 is less than d. Otherwise it returns false.	public class MyDouble{
	public static boolean close(double n1, double n2, double d){
	_____
	_____
	_____
	_____
	}
	}

16. Complete the main method so that it prints GOOD if num1 and num2 are less than 0.1 units apart. Otherwise print BAD.
 	
Call the close method defined in problem 15.
	public class Runner {
	public static void main(String[] args) {
	double num1 = Math.random();
	double num2 = Math.random();
	_____
	_____
	_____
	_____
	_____
	}
	}

17. This code does not compile. Fix it but main method must still call the doThis method.	public class Runner {
	public static void main(String[] args) {
	doThis();
	}
	
	public void doThis(){
	System.out.print("Hey");
	}
	}

18. The Integer class has the following method: public static int parseInt(String s)

Given a String, the parseInt method returns the corresponding int value. For example, if *s* equals "43" then the method returns 43. If the argument cannot be converted to an int then a runtime error occurs. Complete the code snippet below so that the number in str is assigned to num.

```
Scanner scan = new Scanner( System.in );
String str = scan.nextLine();

int num = _____
```

19. Within a class, a static method cannot call a non-static method.	True	False
20. Within a class, an instance method cannot call a static method.	True	False

The Door class compiles as written.

21. If the keyword static was added to the header for the getX method, would the class still compile?

22. If the keyword static was added to the header for the getY method, would the class still compile?

```
public class Door {
        private static int x;
        private int y;

        public Door( int n ){
                x++;
                y = n;
        }

        public int getX(){
                return x;
        }

        public int getY(){
                return y;
        }
}
```

23. What does this display?

```
public class Runner {
        public static void main(String[] args) {
                Door d1 = new Door( 6 );
                Door d2 = new Door( -14 );
                System.out.println( d2.getX() );
                System.out.println( d1.getX() );
                System.out.println( d1.getY() );
                System.out.println( d2.getY() );
        }
}
```

24. Will this compile?

```
public class Paper{
        public void m1(){
                m2();
        }

        public static void m2(){
        }
}
```

25. Will this compile?

26. If the method was changed to an instance method and the instance variable was changed to a class variable, would it compile?

```
public class Scissors{
        private int x;

        public static void m1(){
                x = 5;
        }
}
```

27. All static variables must be public.	True False
28. Class variables are created at the start of the program and exist before any objects of that class are instantiated.	True False

29. Will this compile?	public class Rock{
	public void m1(){
	}
	public static void m2(){
	m1();
	}
	}

Go to http://docs.oracle.com/javase/7/docs/api/ to answer questions 30 to 35.

30. Look up the bitCount method.
If x has a value of 11, what is n?
 	
If x has a value of 16, what is n?
	public class Runner{
	public static void main(String [] args) {
	int x = (int)(17*Math.random());
	int n = Integer.bitCount(x);
	System.out.println(x + ", " + n);
	}
	}

31. Do these lines compile?	int n = "4"; System.out.println(n);
32. Do these lines compile?	int n = Integer.parseInt("4"); System.out.println(n);
33. Do these lines compile? Yes but it generates the following run-time error: java.lang.NumberFormatException	int n = Integer.parseInt("4,000"); System.out.println(n);
34. Do these lines compile?	int n = Double.SIZE; System.out.println(n);
35. How many class/static methods does the Double class have?	

36. What is displayed?	for (int x = 5; x <= 8; x++) {
	for (int y = 0; y <= 4; y++)
	System.out.print("*");
	System.out.println();
	}

37. What is displayed?	for (int a = 1; a <= 3; a++) {
	for (int b = 1; b <= 2; b++)
	System.out.print((a+b) + " ");
	System.out.println();
	}

38. What is displayed?	for (int a = 1; a <= 5; a++) {
	for (int b = 1; b <= a; b++)
	System.out.print(b + " ");
	System.out.println();
	}

39. What is displayed?	for (int g = 6; g > 2; g--) {
	  for (int h = 1; h <= 6; h += 2)
	  System.out.print(g);
	  System.out.println();
	}

40. What is displayed?	for (int a = 10; a <= 50; a += 10) {
	  int b = 2;
	  while (b < a) {
	  System.out.print(b + " ");
	  b = 2 * b;
	  }
	  System.out.println();
	}

41. Complete this method. If *n* is 7, it should print the pattern shown in the top figure. If *n* is 4, it should display something like the bottom figure. Assume that n is greater than zero. You will need to use a loop within a loop.

public void m36(int n) {

_____

_____

_____

_____

_____

_____

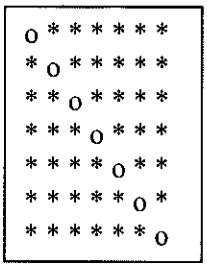
_____

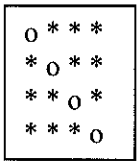
_____

_____

_____

_____





AP CS Unit 5.2: Arrays
Exercises

1. What is displayed?	int [] a = new int[3]; System.out.println(a.length);
2. What is displayed? _____ _____ _____	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]);

3. What is displayed?	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]);

4. What is displayed?	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]);

5. What is displayed?	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);

6. What is displayed?	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);

7. What is displayed?	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?	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.	public class Runner{
	public static void main(String [] args){
	int [] a = { 7, 3, 5, 1 };
	System.out.println(met(a));
	int [] b = { 4, 8, 6, 6, 9 };
	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?

```
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() );
```

```
public class Sack{
    private int[] s;
    private int count;

    public Sack( int n ){
        System.out.println( s );
        s = new int[ n ];
        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?

```
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 ];
    }
}
```

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

```
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?

```
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?

```
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.

```
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.

_____

_____

_____

_____

_____

_____

_____

_____

_____

```
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.

_____

_____

_____

_____

_____

_____

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

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

    public void bark(){
        // implementation code
    }

    // possibly other methods
}
```

22. What is displayed?

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

23. What is displayed?

```
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?

```
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?

```
// 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());
```

```
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 ) {

_____

_____

}
System.out.println( "There are " + num + " even numbers." );
```

27. What is displayed?

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

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

28. What is displayed?

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

29. What is displayed?

```
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( );
}
```

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

```
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( );
}
```

31. What is displayed?

```
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] );
```

32. This code compiles but throws the following runtime exception:
NullPointerException
Why?

```
Cat [][] c = new Cat[2][1];
for ( Cat [] d : c ) {
    for ( Cat e : d )
        System.out.print( e.toString() + " " );
    System.out.println();
}
```

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++ ){

_____

_____
        }
    }

    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:

6	3	5
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 ) {
```