

```
1 public class JavaTracingMethodsDemo1
2 {
3     public static void main(String[] args)
4     {
5         drawTopFloor();
6
7         for (int i = 0; i < 5; i++)
8         {
9             drawMiddleFloor();
10        }
11
12        drawGroundFloor();
13        drawPeople(5);
14    } // end of main
15
16    public static void drawGroundFloor()
17    {
18        System.out.println("*****");
19        System.out.println("*      *");
20        System.out.println("*   ****   *");
21        System.out.println("* * * * *");
22        System.out.println("* * * * *");
23    } // end of drawGroundFloor
24
25    public static void drawTopFloor() {
26        System.out.println("  **");
27        System.out.println(" *  *");
28        System.out.println(" *  *");
29        System.out.println(" *  *");
30        System.out.println("*  *");
31    } // end of drawTopFloor
32
33    public static void drawMiddleFloor() {
34        System.out.println("*****"); System.out.println("*      *");
35        System.out.println("*      *"); System.out.println("*      *");
36        System.out.println("*      *");
37    } // end of drawMiddleFloor
38
39    public static void drawPeople(int num)
40    {
41        for (int i = 0; i < num; i++)
42        {
43            System.out.println(" 0");
44            System.out.println(" ---");
45            System.out.println(" |");
46            System.out.println("/  \");
47            System.out.println();
48        }
49    } // end of drawPeopl
50 }
```

EXTENSION METHODS

5. Define a method `printDigits()` that prints out each of the digits in the argument to the method, separated by commas and on the same line.

```
printDigits(12345)
1, 2, 3, 4, 5
printDigits(1983)
1, 9, 8, 3
```

6. Write the method from number 5, but use a "for" loop or a "while" loop to print the digits (if you didn't already do this in exercise 5)

7. Write a method `getFactors ()` that prints out all of the factors of a number in increasing order

```
getFactors(35)
1 5 7 35
getFactors(7)
1 7
```

```
1 public class JavaTracingMethodsDemo2
2 {
3     public static void main(String[] args)
4     {
5         System.out.println(10 * 10);
6         System.out.println(computeAreaSquare(4));
7         System.out.println(computePerimeterSquare(5));
8         System.out.println(computeAreaTriangle(5, 8));
9         System.out.println(computeAreaRectangle(10, 4));
10        System.out.println(computeAreaRectangleV2(10, 4));
11        int len = 6;
12        int wid = 8;
13        System.out.println(computeAreaRectangle(len, wid));
14        System.out.println(computeAreaRectangle(len + 1, wid * 2));
15    } // end of main
16
17    public static int computeAreaSquare(int width)
18    {
19        return width * width;
20    } // end of computeAreaSquare
21
22    public static int computePerimeterSquare(int width)
23    {
24        return 4 * width;
25    } // end of computePerimeterSquare
26
27    public static double computeAreaTriangle(int base, int height)
28    {
29        return (base * height) / 2;
30    } // end of computeAreaTriangle
31
32    private static int computeAreaRectangle(int length, int width)
33    {
34        return length * width;
35    } // end of computeAreaRectangle
36
37    public static int computeAreaRectangleV2(int length, int width)
38    {
39        int result = 0;
40        result = length * width;
41        return result;
42    } // end of computeAreaRectangleV2
43 }
```


Names:

Vocabulary Focus: Defining and Calling Methods

Study the two class samples and answer the following questions on the methods within each class.

1. Which method in these classes is not defined as public?

computeAreaRectangle ()

2. Is there a non-static method in either of these two classes? How do you know this method is not

static?

computeAreaRectangle v2 ()

3. Which methods in these classes take exactly one parameter? For each method you list, write down the line numbers in the code that indicate that when the method is called, it will need an argument.

computeAreaSquare () # 7, 18

computePerimeterSquare () # 8, 23

main () # 4

4. Which methods in these classes take exactly two parameters? For each method you list, write down the line numbers in the code that indicate that when the method is called, it will need an argument.

computeAreaTriangle () 9, 28

computeAreaRectangle () 10, 33

computeAreaRectangle v2 () 11, 38

5. List the methods in these classes that are 'void' methods? How do we know they are void? What do you think it means for a method to be void?

all methods from Demo 1

no return value or output! → VOID
data

6. List the methods in these classes that will return data? How do we identify a method that has a return value? There are two answers to this question.

all methods from Demo 2 But main ()
→ the method header & the return statement

7. List the method in these classes that returns double data. Is there a logic error in this method that might produce an undesirable output? How would we fix this logic error?

#28, divide by 2.0

8. Describe what's wrong with the method call below? Will this generate a runtime error or a compiler error?

misspelled
computerAreaSquare(4.5);
takes int data

9. There is a variable in one of these methods that is only accessible to that method because it is declared within it? What is the identifier of this variable and which method is it in?

#40, 'result' in computeAreaRectangleV2

10. List several examples of identifiers of 'parameter variables' in these programs.

length, width, height

Java Operators and their Precedence

Now that you've learned how to declare and initialize variables, you probably want to know how to *do something* with them. Learning the operators of the Java programming language is a good place to start. Operators are special symbols that perform specific operations on one, two, or three *operands*, and then return a result.

As we explore the operators of the Java programming language, it may be helpful for you to know ahead of time which operators have the highest precedence. The operators in the following table are listed according to precedence order. The closer to the top of the table an operator appears, the higher its precedence. Operators with higher precedence are evaluated before operators with relatively lower precedence. Operators on the same line have equal precedence. When operators of equal precedence appear in the same expression, a rule must govern which is evaluated first. All binary operators except for the assignment operators are evaluated from left to right; assignment operators are evaluated right to left.

Operator Precedence	
Operators	Precedence
postfix	<code>expr++ expr--</code>
unary	<code>++expr --expr +expr -expr ~ !</code>
multiplicative	<code>* / %</code>
additive	<code>+ -</code>
shift	<code><< >> >>></code>
relational	<code>< > <= >= instanceof</code>
equality	<code>== !=</code>
bitwise AND	<code>&</code>
bitwise exclusive OR	<code>^</code>
bitwise inclusive OR	<code> </code>
logical AND	<code>&&</code>
logical OR	<code> </code>
ternary	<code>? :</code>
assignment	<code>= += -= *= /= %= &= ^= = <<= >>= >>>=</code>

In general-purpose programming, certain operators tend to appear more frequently than others; for example, the assignment operator "=" is far more common than the unsigned right shift operator ">>>".

Logical Operators Worksheet

If $x = -2$, $y = 5$, $z = 0$, and $t = -4$, what is the value of each of the following logical expressions?

1. $x + y < z + 1$
2. $x - 2 * y + y < z * 2 / 3$
3. $3 * y / 4 < 8 \ \&\& \ y \geq 4$
4. $t > 5 \ || \ z < 2$
5. $x * y < 10 \ || \ y * z < 10$
6. $(y + 2) / 3 > 3 \ \&\& \ t < 0$
7. $x * 3 > 0 \ || \ y + 5 / t < 2$
8. $!(x > 0)$
9. $!(x * t < 10) \ || \ y / x * 4 < y * 2$
10. $t > 5 \ || \ z < (y + 5) \ \&\& \ y < 3$
11. $!(4 + 5 * y \geq z - 4) \ \&\& \ (z - 2 < 7)$

Write syntactically correct logical expressions for the following conditions:

1. m is less than 100
2. n is positive and greater than m
3. m is between 5 and 10 (inclusive)
4. k is less than 1 or greater than 2
5. j and k are both negative
6. i is an even number

Logical Operators Worksheet

$x = -2, y = 5, z = 0$, and $t = -4$, what is the value of each of the following logical expressions?

1. $x + y < z + 1$

false

2. $x - 2 * y + y < z * 2 / 3$

true

3. $3 * y / 4 < 8 \ \&\& \ y \geq 4$

true

4. $t > 5 \ || \ z < 2$

true

5. $x * y < 10 \ || \ y * z < 10$

true

6. $(y + 2) / 3 > 3 \ \&\& \ t < 0$

false

7. $x * 3 > 0 \ || \ y + 5 / t < 2$

false

8. $!(x > 0)$

true

9. $!(x * t < 10) \ || \ y / x * 4 < 2$

true

10. $t > 5 \ || \ z < (y + 5) \ \&\& \ < 3$

false

11. $!(4 + 5 * y \geq z - 4) \ \&\& \ -2 < 7$

false

12. Which of the following are syntactically correct logical expressions for the following conditions:

- m is less than 100
- n is positive and greater than m
- n is between 5 and 10 (inclusive)
- n is less than 1 or greater than 2
- m and k are both negative
- k is an even number

$m < 100$
 $n > 0 \ \&\& \ n > m$
 $m >= 5 \ \&\& \ n >= 10$
 $n < 1 \ || \ n > 2$
 $m < 0 \ \&\& \ k < 0$
 $k \% 2 == 0$

— —

54

...nple, the assi

Assessment #3 Practice

Multiple Choice - Five questions covering all concepts covered so far!

skip!
This only works if MAX is multiple of 4

1. What is printed as a result of executing the following code segment?

```

MAX
int k = 0;
while (k < MAX)
{
    k += 4;
}
System.out.println(k);
    
```

- (A) MAX
- (B) MAX - 4
- (C) MAX + 4
- (D) 4
- (E) Nothing - the program goes into an infinite loop

MAX → 12

<i>k</i>	<i><</i>	<i>12?</i>
0		✓
4		✓
8		✓
12		no!

prints 12

only works for multiples of 4

MAX → 20

<i>k</i>	<i><</i>	<i>20?</i>
0		✓
4		✓
8		✓
12		✓
16		✓
20		stop!

prints 20

2. Consider the following method.

```

public int randomPoints(int n)
{
    return (int) (n * Math.random()) + 1;
}
    
```

Which of the following outputs is NOT possible when the statement below is executed?

```

System.out.println(randomPoints(3) + randomPoints(3));
    
```

- (A) 2
- (B) 3
- (C) 4
- (D) 6
- (E) All of the above are possible

*(int) (3 * math.random()) + 1*
produces 1, 2, 3 + *produces 1, 2, 3*
 1+1 → 2
 1+2 → 3
 2+2 → 4
 3+3 → 6

3. What is the output of the following code segment?

```

String name = "DonaldDuck";
int pos = name.indexOf("D", 1);
int pos2 = name.indexOf("al");
int pos3 = name.indexOf("x");
System.out.println(pos + pos2 + pos3);
    
```

- (A) 2
- (B) 3
- (C) 4
- (D) 8
- (E) 9

0 1 2 3 4 5 6 7 8 9
Donald Duck
pos = 6
pos2 = 3
pos3 = -1
6 + 3 + (-1)

4. Given the declaration

```
int p = 5, q = 3;
```

Which of the following expressions evaluates to 7.5?

- I. `(double) (p * q / 2);` $(double)(5 * 3 / 2) \rightarrow (double)(15 / 2) \rightarrow (double)(7.5)$
 II. `(double) p * (double) q / 2;` $5.0 * 3.0 / 2 \rightarrow 15.0 / 2 \rightarrow 7.5$
 III. `(double) p * (double) (q / 2);` $5.0 * (double)(3 / 2)$
 $5.0 * (double)(1) \rightarrow 6.0$

- (A) I only
 (B) II only
 (C) I and II only
 (D) I, II, and III
 (E) None of them

5. Consider the following code segment.

```
int p = 1;
while (p < 5)
{
    int q = 2;
    while (q < 7)
    {
        q += p;
        p++;
        System.out.println(p + " " + q);
    }
}
```

$p < 5$ $q < 7$
 1 ✓ 2 ✓
 2 ← 3 ✓
 3 ← 5 ✓
 4 8

 4 ✓ 2 ✓
 5 ← 6 ✓
 6 11 no!
 6 < 5? no!

prints

2 3
 2 3 3 5 4 8
 2 3 3 5 4 8 5 6
 2 3 3 5 4 8 5 6 6 11

What is the last output when the code segment executes?

- (A) 4 5
 (B) 4 8
 (C) 5 6
 (D) 6 10
 (E) 6 11

Identifying Outputs of "While" Loops

Track the following code segments to determine what they will print.

For problems 1-3, use the following variable declarations:

```
final int MAX = 32;
int num = 15;
```

1. `while (num < MAX)`
`{`
 `num = num++;`
 `System.out.print(num);`
`}`

Output:

15 16 17 18 19 20 21 22 23 24 25 26 27
 28 29 30 31 32

2. `while (num < MAX)`
`{`
 `System.out.println (num);`
 `num *= 2;`
`}`

num < 32?
 15 ✓
 30 ✓
 60 no!

Output:

15
 30

3. `while (num < MAX)`
`{`
 `if (num%3 == 0)`
 `System.out.print(num + " ");`
 `num++;`
`}`

num < 32? %3?
 15 ✓
 16 no
 17 no
 18 ✓
 ↓
 31 no
 32 stop

Output:

15 18 21 24 27 30

4. `int num = 8905;`
`int rev = 0;`
`while (num > 0)`
`{`
 `rev = rev * 10;`
 `rev = num % 10 + rev;`
 `num = num / 10;`
`}`

num rev
 8905 0 0*10
 0 ← 8905%10+0
 5 ✓
 890 5 5*10
 50 ← 890%10+50
 50 ✓
 89
 50 50*10
 500 ← 89%10+500
 89/10 ↓ 509 ✓
 509 509*10
 5090 8%10+5090
 8/10 → 0 5098 ✓

Output:

5098

5. `int num = 5;`
`int dum = 4;`

`while (num > 4 && dum < 10)`
`{`
`num++;`
`dum = num;`
`}`

`System.out.print(num + "-" + dum);`

<code>num > 4</code>	<code>dum < 10</code>	Output:
5 ✓	4 ✓	10-10
6 ✓	6 ✓	
7	7	
8	8	
9	9	
10 ✓	10 no!	

6. `boolean finished = false;`
`int num = 0;`
`while (!finished) {`
`num += 1;`
`if (num % 5 == 0)`
`finished = true;`
`num += 201;`
`}`

`System.out.println(num);`

<code>num</code>	<code>%5 == 0</code>	Output:
0		606
1	φ	
202	φ	
203	φ	
404		
405	✓	→ finished = true
606		stops loop