

1— Which of the following *should* start with a capital letter?

- a. int
- b. string
- c. char
- d. boolean

2— If you wanted to store pi in a data type, which of these would be best?

- a. int
- b. float
- c. char
- d. boolean

3— Which method does Processing run when it executes your program?

- a. void
- b. setup
- c. draw
- d. while

4— Which of the following expressions would return 8.0?

- a. $5 / 2 * 4.0$
- b. $5.0 / 2 * 4$
- c. $5 / 2.0 * 4$
- d. $5.0 / 2.0 * 4.0$

5— Which of the following represents the proper typical structure of a while loop?

- a.

```
while ( test condition, update counter ) {  
    statement;  
    statement;  
    . . . ;  
}
```
- b.

```
while ( initialize counter; test counter; update counter ) {  
    statement;  
    statement;  
    . . . ;  
}
```
- c.

```
if ( while test condition ) {  
    statement;  
    statement;  
    . . . ;  
}
```
- d.

```
while ( test condition ) {  
    statement;  
    statement;  
    . . . ;  
}
```

6— Consider the following code segment,

```
int x = 3; int y = 2;
float z = (float) x / y;
```

(float) is not necessary for this code to compile; **why** is (float) used in this code? What is z with (float) in place? What is z without (float)?

7— Consider the following code segment, which will be used to determine the number of years required for a \$10,000 investment to grow to \$25,000 at a simple interest rate of 3%.

```
int years = 0;
float balance = 10000;
float interest = 0;
while( balance < 25000 ) {
    years++;
    interest = balance * .03;
    <BLANK>
}
System.out.println("Years = " + years);
```

What **line of code** should be placed where <BLANK> is so that the calculation works as intended?

8— Why do the following lines of code not compile? (Explain in one sentence each)

```
float w, x, y and z;
```

```
System.out.println>Hello again!);
```

```
if( x = y ) {
    boolean myBoolean = true;
}
```

9— The method below should print three times the number of "*" as the argument if the number is odd, and double the number of "*" if the number is even, that is:

```
tripleDoubleStar( 1 )    > ***
tripleDoubleStar( 2 )    > ****
tripleDoubleStar( 3 )    > ****
tripleDoubleStar( 4 )    > ****
```

However! It has **three** errors. Correct them!

```
void tripleDoubleStar( int stars ) {
    if( stars % 2 == 1 ) {
        int counter = 0;
        while( counter < stars ) {
            System.Out.Print( "***" );
            counter = counter + 1;
        }
    } else if( stars % 2 == 2 ) {
        while( counter < stars ) {
            System.out.print( "*" );
        }
    }
}
```

10— Write a method that takes three ints as arguments and returns the lowest one.

11— The array question.

Return the sum of the numbers in the array, returning 0 for an empty array.

Except the number 13 is very unlucky, so it does not count towards the sum
and any number immediately following 13 also does not count towards the sum.

{1, 2, 2, 1} > 6

{1, 1} > 2

{1, 2, 2, 1, 13} > 6

{1, 1, 1, 1, 13, 10} > 4

```
public int sum13(int[] nums) {
```

```
}
```