

// 1. **Your challenge: What does this print? Note: there's nothing to fix in this loop, it's OK.**

// How do the print statements differ in that one is inside the loop and the other is outside?

```
int num= 0;
int limit = 5;
while (num <= limit) // do this unless the value in num is greater than the value in limit
{
    num++; // this adds one to num (it's a "postfix increment" operator)
    System.out.println("Value of num is: " + num);
}
System.out.println("Last Value is: " + num); // your brief answer:
```

// 2. What happens here? **How can the code be fixed** to eliminate the problem(s)?

```
int count = 0;
while (count < 5)
{
    count += num; // this is a shortcut assignment operator that means count = count + num;
    num++; // this adds one to num
    System.out.println(count2);
} // your brief answer:
```

// 3. **What needs to be fixed** so that the *while* loop will print the product of the odd integers between 0 and 5, inclusive?

```
int i = 0;
int product = 1;

while (i <= 5)
{
    if (i % 2 != 0) // test if i is odd
        product *= i5; // same as product = product * i5;
    i--; // same as i = i - 1;
}
System.out.println("Product is: " + product); // your brief answer:
```

// 4. **What is wrong** with the following loop? **What is the output after it is fixed?**

```
int sum = 0;
while (i <= 5)
{
    sum += i;
    i++;
}
System.out.println("Sum is: " + sum1); // your brief answer:
```

// 5. What is wrong with this loop? How can you fix it?

```
int i = 10;
while (i >= 0)
{
    if (i % 2 == 0)
        System.out.print(i);
} // your brief answer:
```

// 6. How many times does "When is the class going to end?" print? Note: this loop is OK.

// Note that it's legal to "nest" one loop inside another. The entire "inner" loop executes
// its appropriate number of times each time the outer loop is executed once.

```
int count1 = 1;
while (count1 <= 2)
{
    int count2 = 1;
    while (count2 <= 3)
    {
        System.out.println("When is the class going to end?");
        count2++;
    }
    count1++;
} // your brief answer:
```

// 7. Your challenge: What does this loop print? Note: there's nothing to fix in this loop.

```
for (int k = 0; k < 100; k = k+1)
{
    System.out.println(k);
}
```

Your brief answer:

// 8. How about this one (it's also OK)?

```
for (int k = 0; k < 100; k = k+1)
{
    System.out.println(k++); // k++ returns the original value of k, and then adds 1 to k
}
```

Your brief answer:

// 9. What is wrong with the following *for* loop? What is the output after it is fixed?

```
int k;
for (k = 0; k = 1; k++) // k++ returns the original value of k, and then adds 1 to k
{
    System.out.print(k + " ");
}
```

Your brief answer:

// 10. What is wrong with the following *for* loop? What is the output after it is fixed?

```
for (int k =1, k == 20; k++) {}
```

Your brief answer:

// 11. What does this *do-while* loop print (it's OK as-is)?

```
int k = 3;
do
{
    System.out.print(k);
} while (k != 3);
```

Your brief answer:

// 12. What is wrong with the following *do-while* loop? What is the output after it is fixed?

```
do
{
    System.out.println("This looks correct")
} while {true};
```

Your brief answer:

// 13. Assume String s = "0123456789"; how would you write a *for-each* loop that prints the characters in s, but only those that represent even numbers? A String for-each loop looks like this:

```
for (char c : s.toCharArray()) { ... } // puts each character from s, one at a time, into char c
```

Your answer below: