I can use and interpret definite (for) loops	21. Write a for loop that prints the message “Hi” 1000 times on separate lines.
	
	22. Trace the following code to determine the output.
	
	`double b,q;`
	`double s;`
	`int n;`
	`double qi;`
	`int i;`
	
	`b = 3;`
	`q = 2;`
	`n = 4;`
	
	`s = b;`
	`qi = 1;`
	
	`for (i=1; i<n; i++)`
	`{`
	`qi = qi*q;`
	`s = s+ b*qi;`
	`System.out.println("i = " + i + ", qi = " + qi + ", sum= " + s);`
	`}`
	
	`System.out.println("i = " + i + ", qi = " + qi + ", sum= " + s);`
	
	23. Write code to print out 1, 2, 3, 4, 5

I can use and interpret code that generates a mathematical sequence	24. Write a for loop that prints the squares from 1 to 100 on the same line (no separation)
	
	25. Write a loop to print the integers from 10 to 1 on the same line (separated by spaces)
	
	(Also - see the For Loops Investigation)
	
	26. What sequence of numbers is generated by the code below?
	
	`int a = 0;`
	`int b = 1;`
	
	`for(int n = 1; n <= 6; n++) {`
	`System.out.print(b);`
	
	`int c = b;`
	`b += a;`
	`a = c;`
	`}`

I can use and interpret code that accumulates a value	27. Write a loop to add up all the integer values from -2 to 5
	
	28. Expand the code below so that it reads in 5 numbers from the user and determines the average of these values. (You can assume Scanner is already imported)
	
	`Scanner input = new Scanner(System.in);`
	
	`//your code`
	`System.out.print("Enter a number");`
	`int aNum = input.nextInt();`

I can use and interpret code within Java's scoping rules	29. How could you modify the code so you could use the value in the variable i after the loop terminates?
	
	`for(int i = 0; i < 5; i++) {`
	`System.out.println(i);`
	`}`

I can find the maximum or minimum value	30. Expand the code below so that it reads in 5 numbers from the user and determines which of these values is the largest. (You can assume Scanner is already imported)
	
	`Scanner input = new Scanner(System.in);`
	
	`//your code`
	`System.out.print("Enter a number");`
	`int aNum = input.nextInt();`

I can use and interpret indefinite (while) loops	31. What values are stored in x, y, and q after the following code segment executes?
	
	`int x = 103;`
	`int y = 32;`
	`int q = 0;`
	
	`while(x >= y) {`
	`x = x - y;`
	`q++;`
	`}`

I can identify and fix infinite loops	32. Why does the following code go into an infinite loop?
	
	`Scanner input = new Scanner(System.in);`
	`System.out.print("Enter a number: ");`
	`int next = input.nextInt();`
	`int sum = 0;`
	
	`while(next > 0) {`
	`sum += next;`
	`}`
	`System.out.println(sum);`

	33. Why is the following code an infinite loop?
	
	int count = 0;
	while (count < 100) {
	  System.out.println("count:" + count);
	  count = count * 1;
	}
	
	34. Why is the following code an infinite loop?
	
	int count = 0;
	while (count < 100) {
	  System.out.println("count:" + count);
	}

21.

```
for(int n = 1; n <= 1000; n ++) {
   System.out.println(“Hi”);
}
```

22.

```
i = 1, qi = 2, sum = 9
i = 2, qi = 4, sum = 21
i = 3, qi = 8, sum = 45
i = 4, qi = 8, sum = 45
```

23. One of MANY different solutions:

```
for(int i = 1; i<=5; i++) {
        System.out.print(i);
        if(i != 5) {
                System.out.print(“, ”);
        }
}
```

24.

```
for(int i = 1; i <= 10; i++) {
   System.out.println(i*i);
}
```

25.

```
for(int i = 10; i >= 1; i--) {
   System.out.println(i);
}
```

26.

```
1 1 2 3 5 8
```

27.

```
int sum = 0;
for(int i = -2; i <= 5; i++) {
   sum += i;
}
```

28.

```
int sum = 0;
for(int i = 0; i < 5; i++) {
        System.out.print("Enter a number");
        int aNum = input.nextInt();
        sum += aNum;
}
System.out.println(“Average: ” + sum / 5.0);
```

29.

```
int i; //move the variable declaration before the loop
for(i = 0; i < 5; i++) {
        System.out.println(i);
}
```

30.

```
System.out.print("Enter a number");
int aNum = input.nextInt();
int largest = aNum];
for(int i = 0; i < 5; i++) {
        System.out.print("Enter a number");
        aNum = input.nextInt();
        if(aNum > largest)
                largest = aNum;
}
System.out.println(largest)
```

31. x = 7, y = 32, q = 3

32. A new value is not prompted for in the loop. That means if the user enters in a positive value at the beginning of the program, that the loop condition can never become false.

33. The loop has no code that changes the count variable (multiplying by 1 has no effect)

34. The loop does not change the count variable