

```

    }
}

```

Here is a portion of the output produced by the program:

```

Factors of 2:
Factors of 3:
Factors of 4: 2
Factors of 5:
Factors of 6: 2 3
Factors of 7:
Factors of 8: 2 4
Factors of 9: 3
Factors of 10: 2 5
Factors of 11:
Factors of 12: 2 3 4 6
Factors of 13:
Factors of 14: 2 7
Factors of 15: 3 5
Factors of 16: 2 4 8
Factors of 17:
Factors of 18: 2 3 6 9
Factors of 19:
Factors of 20: 2 4 5 10

```

In the program, the outer loop runs *i* from 2 through 100. The inner loop successively tests all numbers from 2 up to *i*, printing those that evenly divide *i*. Extra challenge: The preceding program can be made more efficient. Can you see how? (Hint: the number of iterations in the inner loop can be reduced.)



Module 3 Mastery Check

1. Write a program that reads characters from the keyboard until a period is received. Have the program count the number of spaces. Report the total at the end of the program.
2. Show the general form of the **if-else-if** ladder.
3. Given

```

if(x < 10)
    if(y > 100) {
        if(!done) x = z;
        else y = z;
    }
else System.out.println("error"); // what if?

```

to what **if** does the last **else** associate?

4. Show the **for** statement for a loop that counts from 1000 to 0 by -2 .
5. Is the following fragment valid?

```
for(int i = 0; i < num; i++)
    sum += i;

count = i;
```

6. Explain what **break** does. Be sure to explain both of its forms.
7. In the following fragment, after the **break** statement executes, what is displayed?

```
for(i = 0; i < 10; i++) {
    while(running) {
        if(x<y) break;
        // ...
    }
    System.out.println("after while");
}
System.out.println("After for");
```

8. What does the following fragment print?

```
for(int i = 0; i<10; i++) {
    System.out.print(i + " ");
    if((i%2) == 0) continue;
    System.out.println();
}
```

9. The iteration expression in a **for** loop need not always alter the loop control variable by a fixed amount. Instead, the loop control variable can change in any arbitrary way. Using this concept, write a program that uses a **for** loop to generate and display the progression 1, 2, 4, 8, 16, 32, and so on.
10. The ASCII lowercase letters are separated from the uppercase letters by 32. Thus, to convert a lowercase letter to uppercase, subtract 32 from it. Use this information to write a program that reads characters from the keyboard. Have it convert all lowercase letters to uppercase, and all uppercase letters to lowercase, displaying the result. Make no changes to any other character. Have the program stop when the user presses period. At the end, have the program display the number of case changes that have taken place.
11. What is an infinite loop?
12. When using **break** with a label, must the label be on a block that contains the **break**?