

Chapter 2 Elementary Programming

1. Valid identifiers: `applet`, `Applet`, `$4`, `apps`, `x`, `y`,
radius
Invalid identifiers: `a++`, `--a`, `4#R`, `#44`, class, public,
int

Keywords:

class, public, int

```
2. double miles = 100;
   final double KILOMETERS_PER_MILE = 1.609;
   double kilometers = KILOMETERS_PER_MILE * miles;
   System.out.println(kilometers);
```

The value of kilometers is 160.9.

3. There are three benefits of using constants: (1) you don't have to repeatedly type the same value; (2) the value can be changed in a single location, if necessary; (3) the program is easy to read.

```
final int SIZE = 20;
```

4.

```
a = 46 / 9; => a = 5
```

```
a = 46 % 9 + 4 * 4 - 2; => a = 1 + 16 - 2 = 15
```

```
a = 45 + 43 % 5 * (23 * 3 % 2); => a = 45 + 3 * (1) = 48
```

```
a %= 3 / a + 3; => a %= 3 + 3; a % = 6 => a = a % 6 = 1;
```

```
d = 4 + d * d + 4; => 4 + 1.0 + 4 = 9.0
```

```
d += 1.5 * 3 + (++a); => d += 4.5 + 2; d += 6.5; => d = 7.5
```

```
d -= 1.5 * 3 + a++; => d -= 4.5 + 1; => d = 1 - 5.5 = -4.5
```

5.

2

2

-4

-4

0

1

6. $(2 + 100) \% 7 = 4$. So it is Thursday.

7. For byte, from -128 to 127, inclusive.
For short, from -32768 to 32767, inclusive.
For int, from -2147483648 to 2147483647, inclusive.

For long, from -9223372036854775808 to 9223372036854775807.

For float, the smallest positive float is 1.40129846432481707e-45 and the largest float is 3.40282346638528860e+38.

For double, the smallest positive double is 4.94065645841246544e-324 and the largest double is 1.79769313486231570e+308d.

8. 25 / 4 is 6. If you want the quotient to be a floating-point number, rewrite it as 25.0 / 4.0, 25.0 / 4, or 25 / 4.0.

9. Yes, the statements are correct. The printout is

```
25 / 4 is 6
25 / 4.0 is 6.25
3 * 2 / 4 is 1
3.0 * 2 / 4 is 1.5
```

10. a. $4.0 / (3.0 * (r + 34)) - 9 * (a + b * c) + (3.0 + d * (2 + a)) / (a + b * d)$

11. $1.0 * m * (r * r)$

12. b and c are true.

13. All.

14.

Line 2: Missing static for the main method.

Line 2: string should be String.

Line 3: i is defined but not initialized before it is used in Line 5.

Line 4: k is an int, cannot assign a double value to k.

Lines 7-8: The string cannot be broken into two lines.

15.

long totalMills = System.currentTimeMillis() returns the milliseconds since Jan 1, 1970.

long totalSeconds = totalMills / 1000 returns the total seconds.

long totalMinutes = totalSeconds / 60 returns the total minutes.

totalMinutes % 60 returns the current minute.

16. Yes. Different types of numeric values can be used in the same computation through numeric conversions referred to as *casting*.

17. The fractional part is truncated. Casting does not change the variable being cast.

18.
f is 12.5
i is 12

19.

```
System.out.println((int)'1');  
System.out.println((int)'A');  
System.out.println((int)'B');  
System.out.println((int)'a');  
System.out.println((int)'b');
```

```
System.out.println((char)40);  
System.out.println((char)59);  
System.out.println((char)79);  
System.out.println((char)85);  
System.out.println((char)90);
```

```
System.out.println((char)0X40);  
System.out.println((char)0X5A);  
System.out.println((char)0X71);  
System.out.println((char)0X72);  
System.out.println((char)0X7A);
```

20. '\u345dE' is wrong. It must have exactly four hex numbers.

21. '\\\'' and '\\\"'

22.

```
i becomes 49, since the ASCII code of '1' is 49;  
j become 99 since (int)'1' is 49 and (int)'2' is 50;  
k becomes 97 since the ASCII code of 'a' is 97;  
c becomes character 'z' since (int) 'z' is 90;
```

23.

```
char c = 'A';  
i = (int)c; // i becomes 65  
  
float f = 1000.34f;  
int i = (int)f; // i becomes 1000  
  
double d = 1000.34;
```

```
int i = (int)d; // i becomes 1000

int i = 97;
char c = (char)i; // c becomes 'a'
```

24.

```
b
c
-2
```

25.

```
System.out.println("1" + 1); => 11
System.out.println('1' + 1); => 50 (since the Unicode for 1 is 49)
System.out.println("1" + 1 + 1); => 111
System.out.println("1" + (1 + 1)); => 12
System.out.println('1' + 1 + 1); => 51
```

26.

```
1 + "Welcome " + 1 + 1 is 1Welcome 11.

1 + "Welcome " + (1 + 1) is 1Welcome 2.

1 + "Welcome " + ('\u0001' + 1) is 1Welcome 2

1 + "Welcome " + 'a' + 1 is 1Welcome a1
```

27. Class names: Capitalize the first letter in each name.
Variables and method names: Lowercase the first word, capitalize the first letter in all subsequent words.
Constants: Capitalize all letters.

28.

```
public class Test {
    /** Main method */
    public static void main(String[] args) {
        // Print a line
        System.out.println("2 % 3 = " + 2 % 3);
    }
}
```

29. Compilation errors are detected by compilers. Runtime errors occur during execution of the program. Logic errors results in incorrect results.

30. The Math class is in the java.lang package. Any class in the java.lang package is automatically imported. So there is no need to import it explicitly.

31. String s = JOptionPane.showInputDialog("Enter an input");

```
32. int i = Integer.parseInt(s);  
    double s = Double.parseDouble(s);
```