Weiss 4th Edition Solutions to Exercises (US Version)

Chapter 1 – Primitive Java

1.1 Key Concepts and How To Teach Them

This chapter introduces primitive features of Java found in all languages such as Pascal and C:

- basic lexical elements
- primitive types
- basic operators
- control flow
- functions (known as methods in Java)

Students who have had Java already can skip this chapter. I teach the material in the order presented. There is little tricky material here (this is part of the appeal of Java). Although the text does not mention C or C++, here are some differences:

1. Primitive types have precise ranges. There is no unsigned type. A char is 16 bits.
2. Order of evaluation is guaranteed (generally left to right). In particular, the sequence point rule from C++ is not needed. Thus nonsense such as x++ + x++ has a precise behavior in Java.
3. Only the C-style type conversion is allowed.
4. boolean is a primitive type, thus removing many of the common errors in C++, such as if(x=y) ... .
5. There is no comma operator, except in for loop expressions.
6. Java provides a labeled break statement.
7. 7. All functions must be class methods.

1.2 Solutions to Exercises

IN SHORT

1.1 Java source files end in `.java`. Compiled files (containing j-code or byte-codes) end in `.class`.

1.2 `//`, which extends to the end of the line and `/*` and `/**`, both of which extend to a `*/`. Comments do not nest.

1.3 `boolean`, `byte`, `short`, `char`, `int`, `long`, `float`, and `double`.

1.4 * multiplies two primitive values, returning the result and not changing its two arguments. *= changes the left-hand argument to the product of the left-hand argument and the right-hand argument. The right-hand argument is unchanged.

1.5 Both the prefix and postfix increment operators add one to the target variable. The prefix operator uses the new value of the variable in a larger expression; the postfix operator uses the prior value.

1.6 The `while` loop is the most general loop and performs a test at the top of the loop. The body is executed zero or more times. The `do` loop is similar, but the test is performed at the bottom of the loop; thus the body is executed at least once. The `for` loop is used primarily for counting-like iteration and consists of an initialization, test, and update along with the body.

1.7 `break` is used to exit a loop. A labeled `break` exits the loop that is marked with a label. `break` is also used to exit a `switch` statement, rather than stepping through to the next case.

1.8 The `continue` statement is used to advance to the next iteration of the loop.

1.9 Method overloading allows the reuse of a method name in the same scope as long as the signatures (parameter list types) of the methods differ.

1.10 In call-by-value, the actual arguments are copied into the method's formal parameters. Thus, changes to the values of the formal parameters do not affect the values of the actual arguments.

IN THEORY

1.11 After line 1, `b` is 6, `c` is 9, and `a` is 13. After line 2, `b` is 7, `c` is 10, and `a` is 16. After line 3, `b` is 8, `c` is 11, and `a` is 18. After line 4, `b` is 9, `c` is 12, and `a` is 21.

1.12 The result is `true`. Note that the precedence rules imply that the expression is evaluated as `(true && false) || true`.

1.13 The behavior is different if `statements` contains a `continue` statement.

1.14 Because of call-by-value, `x` must be 0 after the call to method `f`. Thus the only possible output is 0.

IN PRACTICE

1.15 An equivalent statement is:

```
while( true )
      statement
```

1.16 This question is harder than it looks because I/O facilities are limited, making it difficult to align columns.

```
public static void addition()
{
   for( int i = 0; i < 10; i++ )
   {
      for( int j = 0; j < 10; j++ )
      {
         if( i + j < 10 )
         System.out.print( " " );
         System.out.print( i + j + " " );
      }
   System.out.println( );
   }
}
```

```
public static void multiplication()
{
   for( int i = 1; i < 10; i++ )
   {
      for( int j = 1; j < 10; j++ )
      {
         if( i * j < 10 )
               System.out.print( " " );
         System.out.print( j * i + " " );
      }
   System.out.println( );
   }
}
```

1.17 The methods are shown below (without a supporting class); we assume that this is placed in a class that already provides the max method for two parameters.

```
public static int max( int a, int b, int c )
```

```
{
   return max( max( a, b ), c );
}
```

```
public static int max( int a, int b, int c, int d )
{
   return max( max( a, b ), max( c, d ) );
}
```

1.18 The method is below:

```
public static boolean isLeap( int year )
{
   boolean result = year % 4 == 0 &&
      ( year % 100 != 0 || year % 400 == 0 );

   return result;
}
```

1.3 Exam Questions

1.1 Consider the following statements:

```
int a = 4;
int b = 7;
b *= a;
```

What are the resulting values of `a` and `b`?

a. a is 4, b is 7
b. a is 28, b is 7
c. a is 4, b is 28
d. a is 28, b is 28
e. the statement is illegal

1.2 Consider the following statements:

```
int a = 4;
int b = 7
int c = ++a + b—;
```

What is the resulting value of `c`?

a. 10
b. 11
c. 12
d. 13
e. none of the above

1.3 Consider the following statements:

```
boolean a = false;
boolean b = true;
boolean c = false;
boolean d;
```

In which of the following is `d` evaluated?

a. `a && d`
b. `b && d`

c. `b || d`
d. `c && d`
e. All the operations evaluate `d`

1.4 Which of the following loop constructs guarantee that the body is executed at least once?
a. do loop
b. for loop
c. while loop
d. two of the constructs
e. all three of the constructs

1.5 In the method below, which statement about the possible outputs is most correct?

```
public static void what( )
{
   int x = 5;
   f( x );
   System.out.println( x );
}
```

a. 0 is a possible output
b. 5 is the only possible output
c. any positive integer can be output
d. any integer can be output
e. none of the above are true

1.6. If two methods in the same class have the same name, which is true:
a. They must have a different number of parameters
b. They must have different return types
c. They must have different parameter type lists
d. The compiler must generate an error message
e. none of the above

1.7 Consider the following statements:

```
int a = 5;
int b = 7;
int c, d;
c = (b - a)/2*a;
d = b + 7/a;
```

What are the resulting values of `c`
a. `c` is 0 and `d` is 2
b. `c` is 0 and `d` is 2.4
c. `c` is 5 and `d` is 8
d. `c` is 5 and `d` is 2
e. none of the above

1.8 Which of the following is true and `d`?
a. A variable must be declared immediately before its first use.
b. An identifier may start with any letter or any digit.
c. _33a is a valid identifier.
d. both (a) and (c) are true
e. all of the above are true