Chapter 2 Primitive Data Types and Operations

1. Valid identifiers: `applet`, `Applet`, `$4`, `apps`
   Invalid identifiers: `a++`, `--a`, `4#R`, `#44`

2. ```
   int i = 0;
   long l = 10000;
   float f = 3.4F;
   double d = 34.45;
   char c = '4';
   boolean b = true;
   ```

3. ```
   a = 46/9 = 5
   a = 46%9+4*4-2 = 1+16-2 = 15
   a = 45+43%5*(23*3%2) = 45+3*(1) = 48
   a = 45+45*50%a-- = 45+2250%1 = 45+0 = 45
   a = 45+1+45*50%(++a+1) = 46+2250%3 => 46 + 0 = 46
   d += 1.5*3+(++d); => d += 1.5*3+(++d) => d += 4.5+2.0 => d += 6.5 => d = 7.5
   d -= 1.5*3+d++; => d -= 1.5*3+d++ => d -= 4.5+1.0 => d -= 5.5 => d = 1-5.5 = 4.5
   a %= 3/a+3; => a %= (3/a + 3) => a %= 6 => a = a % 6 =1
   ```

4. 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.

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

6. ASCII code (128 characters) is a subset of Unicode. Unicode (see www.unicode.org for more information) is a 16-bit encoding scheme established by the Unicode Consortium to support the interchange, processing, and display of the written texts of the diverse languages of the world. A Unicode takes 2 bytes, expressed in four hexadecimal numbers that run from \u0000 to \uFFFF. Most computers use ASCII code. Unicode includes ASCII code with \u0000 to \u00FF, corresponding to 128 ASCII characters.

7. The following conversions are not allowed:

   ```
   char c = 'A';
   i = (int)c; // i becomes 65

   boolean b = true;
   i = (int)b; // Not allowed
   ```

```
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'

int i = 1000;
boolean b = (boolean)i; // Not allowed
```

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

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

```
the output for 25/4 is 6;
the output for 25/4.0 is 6.25;
```

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

11. 4.0/(3.0*(r+34)) – 9*(a+b*c) + (3.0+d*(2+a))/(a+b*d)

12. <, <=, ==, !=, >, >=

13.
```
(true) && (3 > 4)
false

!(x > 0) && (x > 0)
false

(x > 0) || (x < 0)
true

(x != 0) || (x == 0)
true

(x >= 0) || (x < 0)
true

(x != 1) == !(x == 1)
true
```

14. `(x > 1) && (x < 100)`

15. `((x > 1) && (x < 100)) || (x < 0)`

16.
```
x > y > 0
incorrect

x = y && y
incorrect

x /= y
correct

x or y
incorrect

x and y
incorrect
```

17. Use // to denote a comment line, and use /* paragraph */ to denote a comment paragraph.

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

19. 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.

20. x is 2.

21. x is 1.

22. The precedence order for boolean operators is &, ^, |, &&, and ||

`true | true && false` is false

`true || true && false` is true

true | true & false is true.

23.

`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

24. b and c are true.

25. The operands are evaluated first and from left to right. So (--i + i + i++) is –1 – 1 –1. i++ return the value of i, then i is incremented by 1. So, in the next println statement i + ++i is 0 + 1.

26. All.

27. a, c, and d.

28. javac

29. java

30. Java interpreter cannot find the .class file. Make sure you placed the .class in the right place, and invoked java command with appropriate package name.

31. The class does not have a main method, or the signature of the main method is incorrect.

32. Missing static for the main method. string should be String. i is defined but not initialized before it is used. k is an int, cannot assign a double value to k. The string cannot be broken into two lines.

33. Change the directory to `c:\smith\homework` and type

```
java csci1301.Test
```