1. { &}
2. If
3. //
4. Space, tab, and newline
5. Keywords
6. Main
7. System.out.print, System.out.printf, and System.out.println
8. False – comments do not force text to be printed on the screen. they are ignored by the compiler
9. True
10. False – java is case sensitive
11. False – The remainder operator % can be used with both integer and floating-point numbers
12. False - \* / % have higher precedence than + -.

if (c < 7); System.out.println("c is less than 7");

error : the semi-colon will terminate if (c < 7)

corrected:

if (c > 7)

System.out.println(“c is less than 7”)

b) if (c => 7) System.out.println("c is equal to or greater than 7");

Error : (=>) is not a valid operator in java

Corrected:

If (c >= 7)

System.out.println(“c is equal to or greater than 7”);

**2.7:**

**a. comments**

**b. if statement**

**c. arithmetic**

**d. division/ $ modulus%**

**e. innermost set**

**f. variable**

**2.9:**

**A. False – operators follow precedence rules, not just left to right.**

**B. True**

**C. False – operator precedence affects evaluation order.**

**D. True**

**2.10:**

a) System.out.printf("x = %d%n", x);

x = 2

b) System.out.printf("Value of %d + %d is %d%n", x, x, (x + x));

The value of 2+2= 4

c) System.out.printf("x =");

Prints “x=” but does not move to a new line

d) System.out.printf("%d = %d%n", (x + y), (y + x));

5 = 5

**2.11:**

a) p=i+j+k+ 7;

Modifies a variable

b) System.out.println("variables whose values are modified");

Does not modify a variable

c) System.out.println("a = 5");

does not modify a variable

d) value = input.nextInt();

Modifies a variable