PKG 04: ELEMENTARY PROGRAMMING 3 and SELECTIONS

ELEMENTARY PROGRAMMING 3, Daniel Liang

Terms:

```
     final double PI = 3.14159;  // Declare a constant
```

constant (or *named constant*) is a permanent data that never changes
final is a Java keyword to declare a constant

Every data type has a range of values. The compiler allocates memory space for each variable or constant according to its data type. Java provides eight primitive data types for numeric values, characters, and Boolean values.

Numerical Data Types:		Storage Size
byte	-2^{7} to $2^{7} - 1$ (-128 to 127)	8-bit signed
short	-2^{15} to $2^{15} - 1$ (-32768 to 32767)	16-bit signed
int	-2^{31} to $2^{31} - 1$ (-2147483648 to 2147483647)	32-bit signed
long	-2^{63} to $2^{63} - 1$ (-9223372036854775808 to 9223372036854775807)	64-bit signed
float	Negative range: -3.4028235E+38 to -1.4E-45 Positive range: 1.4E-45 to 3.4028235E+38	32-bit IEEE 754
double	Negative range: -1.7976931348623157E+308 to -4.9E-324 Positive range: 4.9E-324 to 1.7976931348623157E+308	64-bit IEEE 754

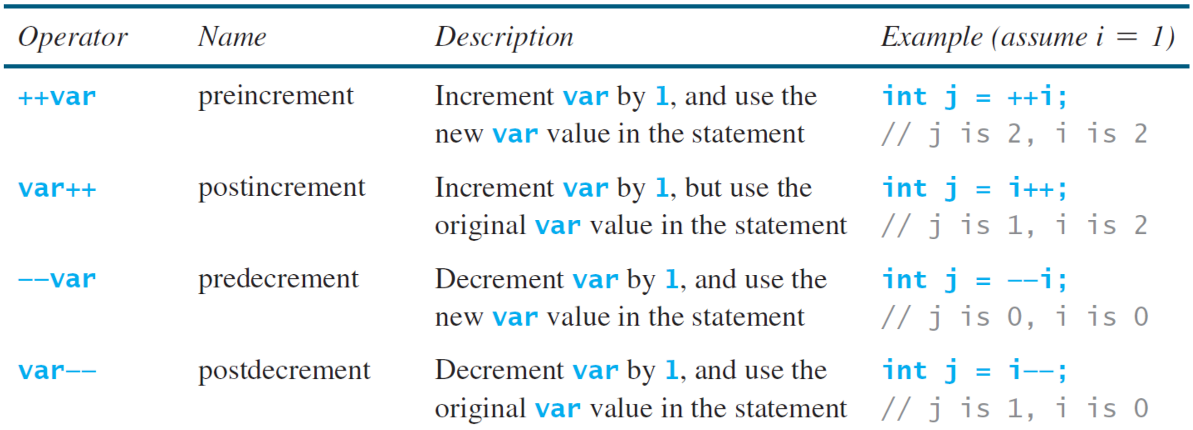
Numeric Operators:

+	Addition	34 + 1	= 35
-	Subtraction	34.0 – 0.1	= 33.9
*	Multiplication	300 * 30	= 9000
/	Division	1.0 / 2.0	= 0.5
%	Remainder	20 % 3	= 2

Augmented Assignment Operators:

+=	Addition	Assignment	i += 8	or	i = i + 8
-=	Subtraction	Assignment	i -= 8	or	i = i – 8
*=	Multiplication	Assignment	i *= 8	or	i = i * 8
/=	Division	Assignment	i /= 8	or	i = i / 8
%=	Remainder	Assignment	i %= 8	or	i = i % 8

Increment and Decrement Operators:



Operator	*Name*	*Description*	*Example (assume i = 1)*
++var	preincrement	Increment var by 1, and use the new var value in the statement	int j = ++i; // j is 2, i is 2
var++	postincrement	Increment var by 1, but use the original var value in the statement	int j = i++; // j is 1, i is 2
--var	predecrement	Decrement var by 1, and use the new var value in the statement	int j = --i; // j is 0, i is 0
var--	postdecrement	Decrement var by 1, and use the original var value in the statement	int j = i--; // j is 1, i is 0

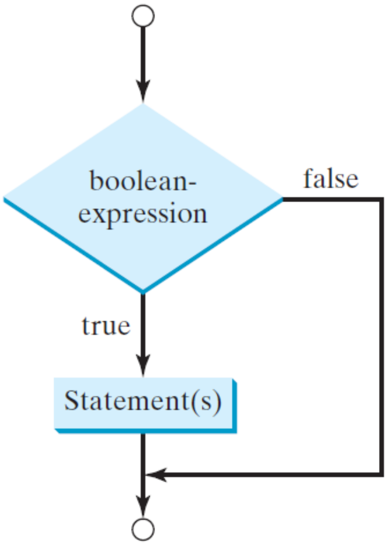
ELEMENTARY PROGRAMMING 3 and SELECTIONS

if STATEMENT, one-way

```
// if statement
if ( 1 < 2 ) {
    System.out.println("Yes, 1 < 2");
}

// another if statement
if ( 3 > 2 ) {
    System.out.println("Yes, 3 > 2");
}
```

BOOLEAN

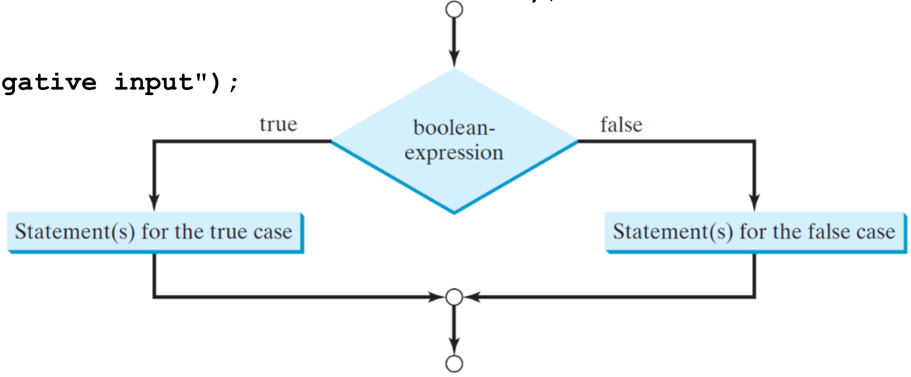
```
boolean c = (1>2);
System.out.println(c); // false

// if statement
if ( c ) {
    System.out.println("Yes, 1 < 2");
}

c = (1<2);
System.out.println(c); // true

c = (1 + 2 == 3);
System.out.println(c); // true

// if statement
if ( c ) {
    System.out.println("Yes, 1 < 2");
}
```



Java Operator	Mathematics Symbol	Name	Example (radius is 5)	Result
`<`	<	less than	`radius < 0`	`false`
`<=`	≤	less than or equal to	`radius <= 0`	`false`
`>`	>	greater than	`radius > 0`	`true`
`>=`	≥	greater than or equal to	`radius >= 0`	`true`
`==`	=	equal to	`radius == 0`	`false`
`!=`	≠	not equal to	`radius != 0`	`true`

if-else STATEMENT, two-way

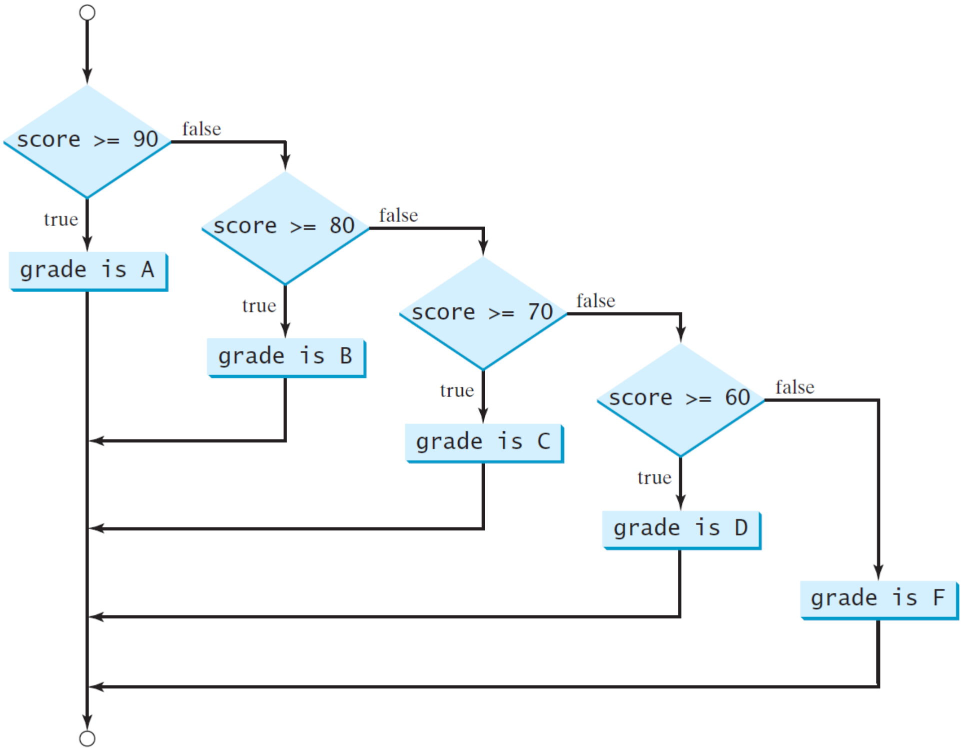
```
if (radius >= 0) {
  area = radius * radius * 3.14159;
      System.out.println("The area for the circle is " + area);
}
else {
      System.out.println("Negative input");
}
```


boolean-expression
true
false
Statement(s) for the true case
Statement(s) for the false case

ELEMENTARY PROGRAMMING 3 and SELECTIONS

if STATEMENT, multi-way

```
if (score >= 90.0) {
    System.out.print("A");
}
else if (score >= 80.0) {
    System.out.print("B");
}
else if (score >= 70.0) {
    System.out.print("C");
}
else if (score >= 60.0) {
    System.out.print("D");
}
else {
    System.out.print("F");
}
```



LOGICAL OPERATORS

Operator	Name	Description
!	not	logical negation
&&	and	logical conjunction
||	or	logical disjunction
^	exclusive or	logical exclusion

```
if (2 > 1 && 2 < 3) {
    System.out.println("Good morning!");
}

if (2 > 1 || 2 > 3) {
    System.out.println("Good morning!");
}

if !(2 < 1) {
    System.out.println("Good morning!");
}
```

Please see lecture slides for:

- Truth Table for not Operator !
- Truth Table for and Operator &&
- Truth Table for or Operator ||
- Truth Table for exclusive or Operator ^
- And more

ELEMENTARY PROGRAMMING 3 and SELECTIONS

1. _______ is the code with natural language mixed with Java code.
a. Java program   b. A Java statement c. Pseudocode d. A flowchart diagram

2. What is the exact output of the following code?

```
  double area = 3.5;
  System.out.print("area");
  System.out.print(area);
```

a. 3.53.5   b. 3.5 3.5   c. area3.5   d. area 3.5

3. Suppose a Scanner object is created as follows:

```
Scanner input = new Scanner(System.in);
```

What method do you use to read a real number?
a. input.nextDouble();   b. input.nextdouble();
c. input.double();   d. input.Double();

4. The following code fragment reads in two numbers:

```
Scanner input = new Scanner(System.in);
int i = input.nextInt();
double d = input.nextDouble();
```

What is the incorrect way to enter these two numbers?
a. Enter an integer, a space, a double value, and then the Enter key.
b. Enter an integer, two spaces, a double value, and then the Enter key.
c. Enter an integer, an Enter key, a double value, and then the Enter key.
d. Enter a numeric value with a decimal point, a space, an integer, and then the Enter key.

5. If you enter 1 2 3, when you run this program, what will be the output?

```
import java.util.Scanner;

public class Test1 {
  public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    System.out.print("Enter three numbers: ");
    double number1 = input.nextDouble();
    double number2 = input.nextDouble();
    double number3 = input.nextDouble();

    double average = (number1 + number2 + number3) / 3;     // Compute average

    System.out.println(average);                            // Display result
  }
}
```

a. 1.0   b. 2.0   c. 3.0   d. 4.0

6. Every letter in a Java keyword is in lowercase?
a. true   b. false

7. Which of the following is a valid identifier?
a. $343   b. class   c. 9X   d. 8+9   e. radius

8. Which of the following are correct names for variables according to Java naming conventions?
a. radius   b. Radius   c. RADIUS   d. findArea   e. FindArea

9. Which of the following are correct ways to declare variables?
a. int length; int width;   b. int length, width;
c. int length; width;   d. int length, int width;

10. ____________ is the Java assignment operator.
a. == b. := c. = d. =:

11. To assign a value 1 to variable x, you write
a. 1 = x; b. x = 1; c. x := 1; d. 1 := x; e. x == 1;

12. Which of the following assignment statements is incorrect?
a. i = j = k = 1; b. i = 1; j = 1; k = 1;
c. i = 1 = j = 1 = k = 1; d. i == j == k == 1;

13. To declare a constant MAX_LENGTH inside a method with value 99.98, you write
a. final MAX_LENGTH = 99.98; b. final float MAX_LENGTH = 99.98;
c. double MAX_LENGTH = 99.98; d. final double MAX_LENGTH = 99.98;

14. Which of the following is a constant, according to Java naming conventions?
a. MAX_VALUE b. Test c. read d. ReadInt e. COUNT

15. To improve readability and maintainability, you should declare _________ instead of using literal values such as 3.14159.
a. variables b. methods c. constants d. classes

16. According to Java naming convention, which of the following names can be variables?
a. FindArea b. findArea
c. totalLength d. TOTAL_LENGTH e. class

17. Which of these data types requires the most amount of memory?
a. long b. int c. short d. byte

18. When assigning a literal to a variable of the byte type, if the literal is too large to be stored as a byte value, it _____________.
a. causes overflow b. causes underflow c. causes no error
d. cannot happen in Java e. receives a compile error

19. What is the result of 45 / 4?
a. 10 b. 11 c. 11.25 d. 12

20. Which of the following expression results in a value 1?
a. 2 % 1 b. 15 % 4 c. 25 % 5 d. 37 % 6

21. 25 % 1 is _____ a. 1 b. 2 c. 3 d. 4 e. 0
22. -25 % 5 is _____ a. 1 b. 2 c. 3 d. 4 e. 0
23. 24 % 5 is _____ a. 1 b. 2 c. 3 d. 4 e. 0
24. -24 % 5 is _____ a. -1 b. -2 c. -3 d. -4 e. 0
25. -24 % -5 is _____ a. 3 b. -3 c. 4 d. -4 e. 0

26. To declare an int variable number with initial value 2, you write
a. int number = 2L; b. int number = 2l; c. int number = 2; d. int number = 2.0;

27. Analyze the following code.

```
public class Test {
  public static void main(String[] args) {
    int month = 09;
    System.out.println("month is " + month);
  }
}
```

a. The program displays month is 09.
b. The program displays month is 9.
c. The program displays month is 9.0.
d. The program has a syntax error, because 09 is an incorrect literal value.

28. Which of the following is incorrect?
a. 1_2 b. 0.4_56 c. 1_200_229 d. _4544

29. Which of the following are the same as 1545.534?
a. 1.545534e+3 b. 0.1545534e+4 c. 1545534.0e-3 d. 154553.4e-2

30. Which of the following is incorrect?
a. int x = 9; b. long x = 9;
c. float x = 1.0; d. double x = 1.0;

31. To add a value 1 to variable x, you write
a. 1 + x = x; b. x += 1;
c. x := 1; d. x = x + 1; e. x = 1 + x;

32. To add number to sum, you write (Note: Java is case-sensitive)
a. number += sum; b. number = sum + number;
c. sum = Number + sum; d. sum += number; e. sum = sum + number;

33. Suppose x is 1. What is x after x += 2?
a. 0 b. 1 c. 2 d. 3 e. 4

34. Suppose x is 1. What is x after x -= 1?
a. 0 b. 1 c. 2 d. -1 e. -2

#
35. What is x after the following statements?

```
int x = 2;
int y = 1;
x *= y + 1;
```

a. x is 1. b. x is 2. c. x is 3. d. x is 4.

36. What is x after the following statements?

```
int x = 1;
x *= x + 1;
```

a. x is 1. b. x is 2. c. x is 3. d. x is 4.

37. Which of the following statements are the same?

```
(A) x -= x + 4
(B) x = x + 4 - x
(C) x = x - (x + 4)
```

a. (A) and (B) are the same b. (A) and (C) are the same
c. (B) and (C) are the same d. (A), (B), and (C) are the same

38. Are the following four statements equivalent?

```
  number += 1;
  number = number + 1;
  number++;
  ++number;
```

a. Yes b. No

39. What is i printed?

```
public class Test {
  public static void main(String[] args) {
    int j = 0;
    int i = ++j + j * 5;

    System.out.println("What is i? " + i);
  }
}
```

a. 0 b. 1 c. 5 d. 6

40. What is i printed in the following code?

```
public class Test {
  public static void main(String[] args) {
    int j = 0;
    int i = j++ + j * 5;

    System.out.println("What is i? " + i);
  }
}
```

a. 0   b. 1   c. 5   d. 6

41. What is y displayed in the following code?

```
public class Test {
  public static void main(String[] args) {
    int x = 1;
    int y = x++ + x;
    System.out.println("y is " + y);
  }
}
```

a. y is 1.   b. y is 2.   c. y is 3.   d. y is 4.

42. What is y displayed?

```
public class Test {
  public static void main(String[] args) {
    int x = 1;
    int y = x + x++;
    System.out.println("y is " + y);
  }
}
```

a. y is 1.   b. y is 2.   c. y is 3.   d. y is 4.

43. To assign a double variable d to a float variable x, you write

a. x = (long)d   b. x = (int)d;
c. x = d;   d. x = (float)d;

44. Which of the following expressions will yield 0.5?

a. 1 / 2   b. 1.0 / 2   c. (double) (1 / 2)
d. (double) 1 / 2   e. 1 / 2.0

45. What is the output of the following code:

```
double x = 5.5;
int y = (int)x;
System.out.println("x is " + x + " and y is " + y);
```

a. x is 5 and y is 6   b. x is 6.0 and y is 6.0   c. x is 6 and y is 6
d. x is 5.5 and y is 5   e. x is 5.5 and y is 5.0

46. Which of the following assignment statements is illegal?

a. float f = -34;   b. int t = 23;   c. short s = 10;
d. int t = (int)false;   e. int t = 4.5;

47. What is the value of (double)5/2?   a. 2   b. 2.5   c. 3   d. 2.0   e. 3.0

48. What is the value of (double)(5/2)?   a. 2   b. 2.5   c. 3   d. 2.0   e. 3.0

49. Which of the following expression results in 45.37?

a. (int)(45.378 * 100) / 100   b. (int)(45.378 * 100) / 100.0
c. (int)(45.378 * 100 / 100)   d. (int)(45.378) * 100 / 100.0

50. The expression (int)(76.0252175 * 100) / 100 evaluates to _________.

a. 76.02   b. 76   c. 76.0252175   d. 76.03

51. If you attempt to add an int, a byte, a long, and a double, the result will be a __________ value.   a. byte   b. int   c. long   d. double

```
import java.text.DecimalFormat;
import java.util.Calendar;
import java.util.Locale;

public class NumberFormatting {

    public static void main(String[] args) {

        // 16 D and 8 F
        System.out.println("\n########## For ASMT 1 ##########");
        System.out.println("1 ASMT1 => 1.0 / 3.0 is: " + 1.0 / 3.0);
                                                            // --> 0.3333333333333333
        System.out.println("2 ASMT1 => 1.0F / 3.0F is: " + 1.0F / 3.0F);
                                                            // -->  0.33333334
        // Not for ASMT 1
        System.out.println("\n\n########## Not for ASMT 1 ##########");
        double d;
        d = 1.0 / 3.0;
        System.out.println("1 => 100.0 / 3.0 is: " + 100.0 / 3.0);
                                                            // --> 33.333333333333336
        System.out.format("2 =>    1.0F / 3.0F is: %028.24f%n", d);
                                                  // --> 000.333333333333333300000000
        d = 1000.0 / 3.0;
        System.out.format("3 => 1000.0F / 3.0F is: %028.24f%n", d);
                                                  // --> 333.333333333333300000000000
        // format or printf
        System.out.println("\n\n########## Some Numeric Formatting ##########");
        long n = 461012;
        System.out.format("01 => %d%n", n);     //  -->  "461012"
        System.out.format("02 => %08d%n", n);   //  -->  "00461012"
        System.out.format("03 => %+8d%n", n);   //  -->  " +461012"
        System.out.format("04 => %,8d%n", n);   // -->  " 461,012
        System.out.format("05 => %-,8d%n", n);  // -->  " 461,012
        System.out.format("06 => %+,8d%n%n", n); //  -->  "+461,012

        double pi = Math.PI;

        System.out.format("07 => %f%n", pi);      // -->  "3.141593"
        System.out.format("08 => %.3f%n", pi);    // -->  "3.142"
        System.out.format("09 => %10.3f%n", pi);  // -->  "     3.142"
        System.out.format("10 => %-10.3f%n", pi); // -->  "3.142"
        System.out.print("11 => ");
        System.out.format(Locale.FRANCE, "%-10.4f%n%n", pi); // -->  "3,1416"

        Calendar c = Calendar.getInstance();
        System.out.format("12 => %tB %te, %tY%n", c, c, c); // --> "September 4, 2019"
        System.out.format("13 => %tl:%tM %tp%n", c, c, c);  // --> "2:34 am"
        System.out.format("14 => %tD%n", c);                 // -->  "05/29/06"

        // Custom format
        System.out.println("\n\n########## And some more ##########");
        customFormat("###,###.###", 123456.789);  // --> 123,456.789
        customFormat("###.##", 123456.789);       // --> 123456.79
        customFormat("000000.000", 123.78);       // --> 000123.780
        customFormat("$###,###.###", 12345.67);   // --> $12,345.67
    }

    static public void customFormat(String pattern, double value) {
        DecimalFormat myFormatter = new DecimalFormat(pattern);
        String output = myFormatter.format(value);
        System.out.println(output);
    }
}
```