

Exercises: 3, 6, 8, 12, and 34; Web exercises: 11, 15, 50, 51, 53, and 56.

Ex 3 What (if anything) is wrong with each of the following statements?

- a. `if (a > b) then c = 0;`
- b. `if a > b { c = 0; }`
- c. `if (a > b) c = 0;`
- d. `if (a > b) c = 0 else b = 0;`

R. a, b, c are wrong

ex 6: Suppose that `i` and `j` are both of type `int`. What is the value of `j` after each of the following statements is executed?

- a. `for (i = 0, j = 0; i < 10; i++) j += i;`
- b. `for (i = 0, j = 1; i < 10; i++) j += j;`
- c. `for (j = 0; j < 10; j++) j += j;`
- d. `for (i = 0, j = 0; i < 10; i++) j += j++;`

R.

i - j   a  
0 - 0  
1 - 1  
2 - 3  
3 - 6  
4 - 10  
5 - 15  
6 - 21  
7 - 28  
8 - 36  
9 - 45

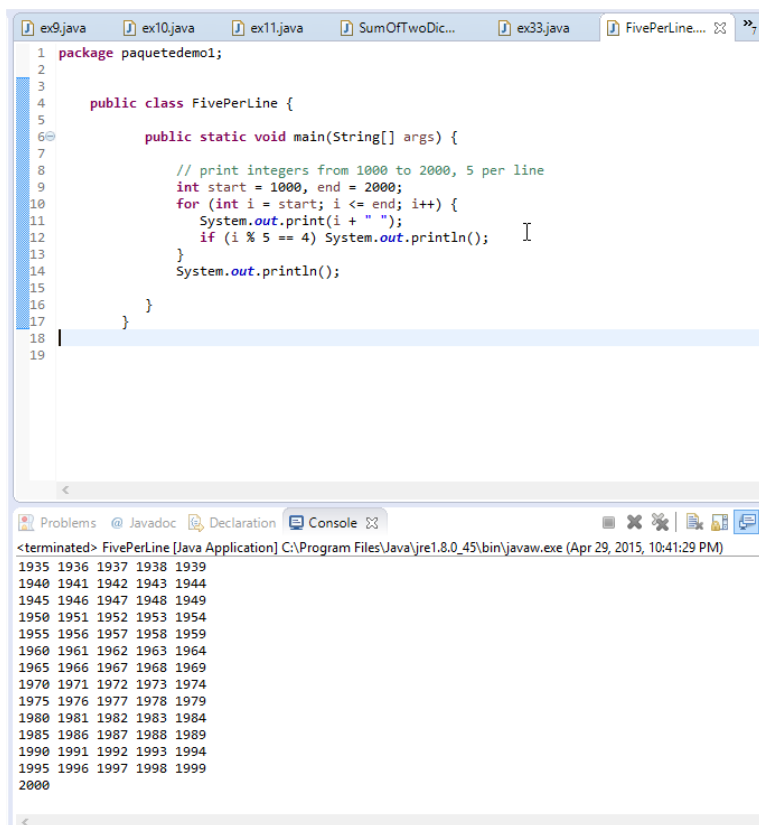
i - j   b  
0 - 2  
1 - 4  
2 - 8  
3 - 16  
4 - 32  
5 - 64  
6 - 128  
7 - 256  
8 - 512  
9 - 1024

```

i - j  c
10 - 0
10 - 2
10 - 6
10 - 14
-
i - j  d
0 - 0
1 - 0
2 - 0
3 - 0
4 - 0
5 - 0
6 - 0
7 - 0
8 - 0
9 - 0

```

Ex8 Write a program [FivePerLine.java](#) that, using one `for` loop and one `if` statement, prints the integers from 1000 to 2000 with five integers per line. *Hint:* use the `%` operator.



```

1 package paquetedemo1;
2
3
4 public class FivePerLine {
5
6     public static void main(String[] args) {
7
8         // print integers from 1000 to 2000, 5 per line
9         int start = 1000, end = 2000;
10        for (int i = start; i <= end; i++) {
11            System.out.print(i + " ");
12            if (i % 5 == 4) System.out.println();
13        }
14        System.out.println();
15    }
16 }
17
18
19

```

```

<terminated> FivePerLine [Java Application] C:\Program Files\Java\jre1.8.0_45\bin\javaw.exe (Apr 29, 2015, 10:41:29 PM)
1935 1936 1937 1938 1939
1940 1941 1942 1943 1944
1945 1946 1947 1948 1949
1950 1951 1952 1953 1954
1955 1956 1957 1958 1959
1960 1961 1962 1963 1964
1965 1966 1967 1968 1969
1970 1971 1972 1973 1974
1975 1976 1977 1978 1979
1980 1981 1982 1983 1984
1985 1986 1987 1988 1989
1990 1991 1992 1993 1994
1995 1996 1997 1998 1999
2000

```

1. Ex 12 What is the value of `m` and `n` after executing the [following code](#)?

```

int n = 123456789;
int m = 0;
while (n != 0) {

```

```

        m = (10 * m) + (n % 10);
        n = n / 10;
    }

```

R. 987654321

Ex 34 **Calendar**. Write a program `Calendar` that takes two command line arguments `m` and `y` and prints out the monthly calendar for the `m`th month of year `y`. For example, your output for `Calendar 2 2009` should be

The screenshot shows an IDE with the `Calendar.java` file open. The code implements a program that takes two command-line arguments, `m` (month) and `y` (year), and prints a calendar for the specified month and year. The code includes methods for calculating the day of the week for a given date, checking for leap years, and printing the calendar grid.

```

1 package paquetedemo1;
2
3
4 public class Calendar {
5
6     public static int day(int M, int D, int Y) {
7         int y = Y - (14 - M) / 12;
8         int x = y + y/4 - y/100 + y/400;
9         int m = M + 12 * ((14 - M) / 12) - 2;
10        int d = (D + x + (31*m)/12) % 7;
11        return d;
12    }
13
14    public static boolean isLeapYear(int year) {
15        if ((year % 4 == 0) && (year % 100 != 0)) return true;
16        if (year % 400 == 0) return true;
17        return false;
18    }
19
20    public static void main(String[] args) {
21        int M = Integer.parseInt(args[0]);
22        int Y = Integer.parseInt(args[1]);
23
24        String[] months = {"", "January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"};
25        int[] days = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
26
27        if (M == 2 && isLeapYear(Y)) days[M] = 29;
28
29        System.out.println("    " + months[M] + "    " + Y);
30        System.out.println(" S M Tu W Th F S");
31
32        int d = day(M, 1, Y);
33
34        // print the calendar
35        for (int i = 0; i < d; i++)
36            System.out.print("    ");
37        for (int i = 1; i <= days[M]; i++) {
38            System.out.printf("%2d ", i);
39            if (((i + d) % 7 == 0) || (i == days[M])) System.out.println();
40        }
41
42    }
43 }
44

```

The console output shows the calendar for February 2009:

```

<terminated> Calendar [Java Application] C:\Program Files\Java\jre1.8.0_45\bin\javaw.exe (Apr 29, 2015, 10:56:41 PM)
February 2009
S M Tu W Th F S
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28

```

Web

1. Ex 11 What is wrong with the following code fragment?

```
double x = -32.2;
boolean isPositive = (x > 0);
if (isPositive == true) System.out.println(x + " is positive");
else                      System.out.println(x + " is not positive");
```

R. The condition into if require change = to ==

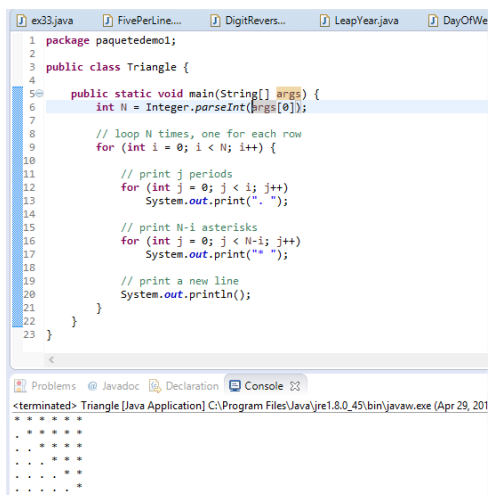
Ex 15 What does the following program do?

```
public static void main(String[] args) {
    int N = Integer.parseInt(args[0]);
    int x = 1;
    while (N >= 1) {
        System.out.println(x);
        x = 2 * x;
        N = N / 2;
    }
}
```

Increase \* 2 the value while this are low than initial value setter

Example  $N = 50$  result 1 2 4 8 16 32.

Ex 50 Write a program [Triangle.java](#) that takes a command-line argument N and prints an N-by-N triangular pattern like the one below.



Ex 51 Write a program [Ex.java](#) that takes a command-line argument N and prints a  $(2N + 1)$ -by- $(2N + 1)$  ex like the one below. Use two `for` loops and one `if-else` statement.

```
1 package paquetedemo1;
2
3 public class Ex {
4
5     public static void main(String[] args) {
6         int N = 6;
7
8         for (int i = -N; i <= N; i++) {
9             for (int j = -N; j <= N; j++) {
10                 if ((i == -j) || (i == j)) System.out.print("* ");
11                 else System.out.print(". ");
12             }
13             System.out.println();
14         }
15     }
16 }
```

I

< Problems @ Javadoc Declaration Console <

<terminated> Ex [Java Application] C:\Program Files\Java\jre1.8.0\_45\bin\javaw.exe (Apr 29, 2015, 11:11:25 PM)

```
* . . . . . *
. * . . . . *
. . * . . . *
. . . * . . *
. . . . * . *
. . . . * . *
. . . . * . *
. . . * . . *
. . * . . . *
. * . . . . *
* . . . . . *
```

Ex 53 Write a program [Diamond.java](#) that takes a command-line argument N and prints a  $(2N + 1)$ -by- $(2N + 1)$  diamond like the one below.

```
DigitRevers... LeapYear.java DayOfWeek.java Calendar.java Triangle.java  
1 package paquetedemo1;  
2  
3 public class Diamond {  
4  
5     public static void main(String[] args) {  
6         int N = 6;  
7  
8         for (int i = -N; i <= N; i++) {  
9             for (int j = -N; j <= N; j++) {  
10                 if (Math.abs(i) + Math.abs(j) <= N) System.out.print("** ");  
11                 else System.out.print(" ");  
12             }  
13             System.out.println();  
14         }  
15     }  
16 }  
17  
18  
19
```

<terminated> Diamond [Java Application] C:\Program Files\Java\jre1.8.0\_45\bin\javaw.exe (Apr 29, 2015, 11:13:54 PM)

```
. . . * . . .  
. . . ** . . .  
. . . *** . . .  
. . . **** . . .  
. . . ***** . . .  
*****  
*****  
*****  
****  
***  
**  
*
```