

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

if (a > b) then c = 0; word then

if a > b { c = 0; } () missing on condition

if (a > b) c = 0; Nothing wrong

if (a > b) c = 0 else b = 0; missing ; before else

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?

for (i = 0, j = 0; i < 10; i++) j += i; **45**

for (i = 0, j = 1; i < 10; i++) j += j; **1024**

for (j = 0; j < 10; j++) j += j; **15**

for (i = 0, j = 0; i < 10; i++) j += j++; **0**

The screenshot shows an IDE with a project named 'root' containing several Java files. The file 'Section136.java' is open, showing the following code:

```
1 public class Section136 {
2
3     public static void main(String []args){
4         int i, j;
5
6         for (i = 0, j = 0; i < 10; i++) j += i;
7         System.out.println(j);
8
9         for (i = 0, j = 1; i < 10; i++) j += j;
10        System.out.println(j);
11
12        for (j = 0; j < 10; j++) j += j;
13        System.out.println(j);
14
15        for (i = 0, j = 0; i < 10; i++) j += j++;
16        System.out.println(j);
17    }
18 }
19
20 }
```

The output of the program is shown in the terminal window at the bottom:

```
sh-4.3# javac Section136.java && java Section136
45
1024
15
0
sh-4.3#
```

8. 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.

The screenshot shows the CodingGround IDE interface. The top bar includes the logo, version (JDK 1.7.0), and a system icon. The left sidebar displays a project tree with files like Diamond.class, Diamond.java, Ex.class, Ex.java, FivePerLine.class, FivePerLine.java, HelloWorld.java, Newfile.java, Section1315.class, Section1315.java, Triangle.class, and Triangle.java. The main editor window shows the code for FivePerLine.java, which is a public class with a main method that prints integers from 1000 to 2000, 5 per line, with a space after each number. The output window at the bottom shows the execution result, displaying numbers from 1850 to 2000 in groups of five, with a space after each number.

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

Terminal output:

```
1850 1851 1852 1853 1854
1855 1856 1857 1858 1859
1860 1861 1862 1863 1864
1865 1866 1867 1868 1869
1870 1871 1872 1873 1874
1875 1876 1877 1878 1879
1880 1881 1882 1883 1884
1885 1886 1887 1888 1889
1890 1891 1892 1893 1894
1895 1896 1897 1898 1899
1900 1901 1902 1903 1904
1905 1906 1907 1908 1909
1910 1911 1912 1913 1914
1915 1916 1917 1918 1919
1920 1921 1922 1923 1924
1925 1926 1927 1928 1929
1930 1931 1932 1933 1934
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
```

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;
```

```
}
```

M will be 9876543210 and n equals to 0

```
New Project | Compile | Execute | Ex.java x | Triangle.java x | Sect
root
├── Diamond.class
├── Diamond.java
├── Ex.class
├── Ex.java
├── FivePerLine.class
├── FivePerLine.java
├── HelloWorld.java
├── Newfile.java
├── Section1315.class
├── Section1315.java
├── Triangle.class
├── Triangle.java
└── Section1312.java

1 public class Section1312{
2
3 public static void main(String []args){
4     int n = 123456789;
5     int m = 0;
6     while (n != 0) {
7         m = (10 * m) + (n % 10);
8         n = n / 10;
9     }
10    System.out.println(m + " " + n);
11 }
12 }
```

```
Terminal
sh-4.3# javac Section1312.java && java Section1312
987654321 0
sh-4.3#
```

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

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

Hint: see programs LeapYear.java and DayOfWeek.java.

Web Exercises

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");
```

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);

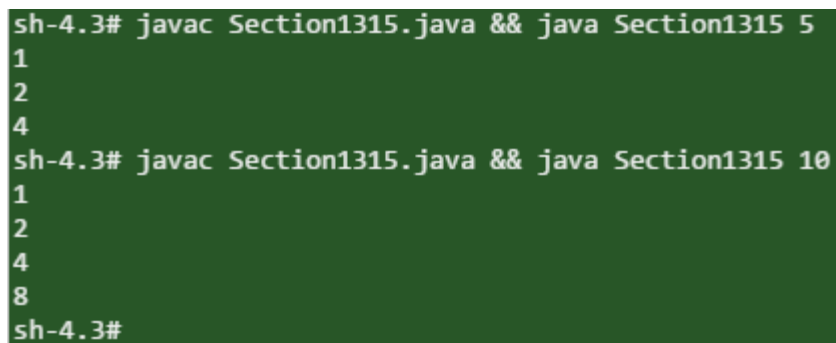
        x = 2 * x;

        N = N / 2;

    }

}
```

It prints power of two less than the N number (parameter), examples:



```
sh-4.3# javac Section1315.java && java Section1315 5
1
2
4
sh-4.3# javac Section1315.java && java Section1315 10
1
2
4
8
sh-4.3#
```

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.

```
* * * * *
```

```

. * * * *
.
. . * * *
. .
. . . * *
. . .
. . . . *
. . . .
. . . . . *
. . . . .

```

The screenshot shows an IDE with a project named 'root' containing several files. The 'Triangle.java' file is selected and its code is visible in the editor. The code defines a public class 'Triangle' with a 'main' method that takes an array of strings 'args'. It parses the first argument as an integer 'N'. It then uses two nested loops: an outer loop for 'i' from 0 to N-1, and an inner loop for 'j' from 0 to i. The inner loop prints spaces, and the outer loop prints asterisks. The terminal output shows the execution of 'java Triangle 5', which produces a 5x5 triangle of asterisks and spaces, matching the pattern shown in the first block.

```

1 public class Triangle {
2
3     public static void main(String[] args) {
4         int N = Integer.parseInt(args[0]);
5
6         // loop N times, one for each row
7         for (int i = 0; i < N; i++) {
8
9             // print j periods
10            for (int j = 0; j < i; j++)
11                System.out.print(" ");
12
13            // print N-i asterisks
14            for (int j = 0; j < N-i; j++)
15                System.out.print("* ");
16
17        }
18    }
19 }

```

```

sh-4.3# javac Triangle.java && java Triangle 5
. * * * *
.
. . * * *
. .
. . . * *
. . .
. . . . *
. . . .
. . . . . *
. . . . .
sh-4.3#

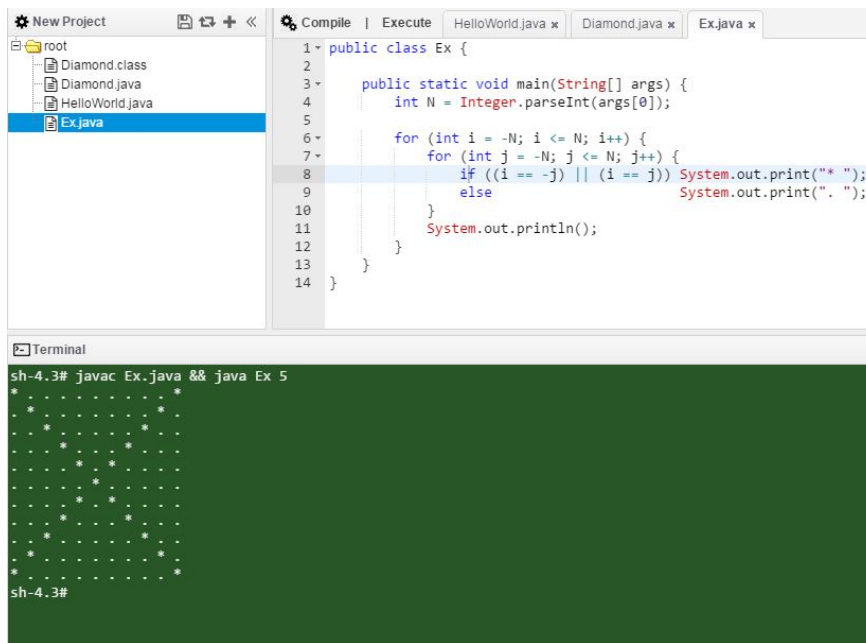
```

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.

```

* . . . . *
.
. * . . *
. .
. . * * .
. .
. . . * .
. .
. . * * .
. .
. * . . *
. .
* . . . . *
.

```



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.

% java Diamond 4

```

    *
  . . .
    * * *
  . . .
    * * * *
  . .
    * * * * *
  .
    * * * * * *
  .
    * * * * *
  .
    * * * *
  . .
    * * *
  . . .
    *
  . . .

```

The screenshot shows an IDE with a project named 'New Project'. The file explorer on the left shows 'root' containing 'HelloWorld.java' and 'Diamond.java'. The main editor displays the code for 'Diamond.java':

```
1 public class Diamond {
2
3     public static void main(String[] args) {
4         int N = Integer.parseInt(args[0]);
5
6         for (int i = -N; i <= N; i++) {
7             for (int j = -N; j <= N; j++) {
8                 if (Math.abs(i) + Math.abs(j) <= N) System.out.print("* ");
9                 else System.out.print(" ");
10            }
11            System.out.println();
12        }
13    }
14 }
15
```

The terminal at the bottom shows the command: `sh-4.3# javac Diamond.java && java Diamond 4`. The output is a diamond shape of asterisks:

```

 *
* *
* * *
* * * *
* * * *
* * * *
* * * *
* * * *
* * *
* *
*

```

56. Seasons. Write a program Season.java that takes two command line integers M and D and prints the season corresponding to month M (1 = January, 12 = December) and day D in the northern hemisphere. Use the following table

SEASON	FROM	TO
Spring	March 21	June 20
Summer	June 21	September 22
Fall	September 23	December 21
Winter	December 21	March 20