

Mathematical functions, Characters, and strings

Chapter 4



September 4, 2019

Thomas J. Short

Cis 260-01

**Checkpoints: Chapter 4**

**2.1**

Evaluate the following method calls:

(a) Math. sqrt (4) is **2.0**

(b) Math. sin (2 \* Math. PI) is **0**

(c) Math. cos (2 \* Math. PI) is **1**

(d) Math. pow (2, 2) is **4.0**

(e) Math. log (Math. E) is **1.0**

(f) Math. exp (1) is **2.718**

(g) Math. max (2, Math. min (3, 4)) is **3**

(h) Math. rint (-2.5) is **-2.0**

(i) Math. ceil (-2.5) is **-2.0**

(j) Math. floor (-2.5) is **-3.0**

(k) Math. round (-2.5f) is **-2**

(l) Math. round (-2.5) is **-2**

(m) Math. rint (2.5) is **2.0**

(n) Math. ceil (2.5) is **3.0**

(o) Math. floor (2.5) is **2.0**

(p) Math. round (2.5f) is **3**

(q) Math. round (2.5) is **3**

(r) Math. round(Math. abs (-2.5)) is **3**

**2.5**

Write an expression that obtains a random integer between 34 and 55. Write an expression that obtains a random integer between 0 and 999. Write an expression that obtains a random number between 5.5 and 55.5.

**(a) 34 + (int) (Math. random() \* (21))**

**(b) (int) (Math. random() \* 1000)**

**(c) 5.5 + (Math. random() \* (50))**

**2.6**

Why does the Math class not need to be imported?

**The math class does not need to be imported because it resides in the java.lang package which is automatically imported. There is no reason to explicitly import the math class**

**3.1**

Use print statements to find out the ASCII code for '1', 'A', 'B', 'a', and 'b'. Use print statements to find out the character for the decimal codes 40, 59, 79, 85, and 90. Use print statements to find out the character for the hexadecimal code 40, 5A, 71, 72, and 7A.

System.out.println((int)'1'); is **49**

System.out.println((int)'A'); is **65**

System.out.println((int)'B'); is **66**

System.out.println((int)'a'); is **97**

System.out.println((int)'b'); is **98**

System.out.println((char)40); is **(**

System.out.println((char)59); is **;**

System.out.println((char)79); is capital **O**

System.out.println((char)85); is capital **U**

System.out.println((char)90); is Capital **Z**

System.out.println((char)0X40); is **@**

System.out.println((char)0X5A); is capital **Z**

System.out.println((char)0X71); is lowercase **q**

System.out.println((char)0X72); is lowercase **r**

System.out.println((char)0X7A); is lowercase **z**

**3.3**

How do you display the characters \ and "?

**‘\\’ or ‘/u005C’ or ‘92’ are all valid code for backslash**

**‘/”’ or ‘/u0022’ or ‘34’ are all valid code for double Quote**

**3.4**

Evaluate the following:

int i = ‘1’; is **49**, since 0 is 48

int j = ‘1’ + ‘2’ \* (‘4’ – ‘3’) + ‘b’ / ‘a’; is **100**

int k = ‘a’; is **97** since lowercase letters start at 97 in hexadecimal

char c = 90; is Capital **Z** because capital letters end with 90

**3.6**

Show the output of the following program:

public class Test {

public static void main(String[] args) {

char x = 'a';

char y = 'c';

System.out.Println(++x); // **b** because a is assigned //to x and x increments before we us it

System.out.println(y++); // **c** because y will not //increment until after we use it

System.out.println(x - y); // **-2**  letter a increments before we use it so x is 98 and b changes from 99 to 100 because we are using it after so 98 – 100 is -2

}

}

**4.1**

Suppose that s1, s2, and s3 are three strings, given as follows:

String s1 = "Welcome to Java";

String s2 = "Programming is fun";

String s3 = "Welcome to Java";

What are the results of the following expressions?

(a) s1 == s2

(b) s2 == s3

(c) s1.equals(s2)

(d) s1.equals(s3)

(e) s1.compareTo(s2)

(f) s2.compareTo(s3)

(g) s2.compareTo(s2)

(h) s1.charAt(0)

(i) s1.indexOf('j')

(j) s1.indexOf("to")

(k) s1.lastIndexOf('a')

(l) s1.lastIndexOf("o", 15)

(m) s1.length()

(n) s1.substring(5)

(o) s1.substring(5, 11)

(p) s1.startsWith("Wel")

(q) s1.endsWith("Java")

(r) s1.toLowerCase()

(s) s1.toUpperCase()

(t) s1.concat(s2)

(u) s1.contains(s2)

(v) "\t Wel \t".trim()

(a)**false**

(b)**false**

(c)**false**

(d)**true**

(e)a **positive number 3**

(f)a **negative number -3**

(g)**0**

(h)**W**

(i)**-1**

(j)**8**

(k)**14**

(l)**9**

(m)**15**

(n)**me to Java**

(o)**me to**

(p)**true**

(q)**true**

(r)**welcome to java**

(s)**WELCOME TO JAVA**

(t)**Welcome to JavaProgramming is fun**

(u)**false**

(v)**Wel**

**4.5**

Let s1 be " Welcome " and s2 be " welcome ". Write the code for the following statements:

1. Check whether s1 is equal to s2 and assign the result to a boolean variable isEqual.

**boolean isEqual = s1.equals(s.2)**

1. Check whether s1 is equal to s2, ignoring case, and assign the result to a Boolean variable isEqual.

**boolean is Equal = s1.equalsIgnoreCase (s2);**

1. Compare s1 with s2 and assign the result to an int variable x.

**int x = s1.compareTo(s.2);**

1. Compare s1 with s2, ignoring case, and assign the result to an int variable x.

**Int x = s1.compareToIgnoreCase(s2);**

1. Check whether s1 has the prefix AAA and assign the result to a Boolean variable b.

**Boolean b = s1.startWith(AAA);**

1. Check whether s1 has the suffix AAA and assign the result to a Boolean variable b.

**Boolean b = s1.suffix(AAA);**

1. Assign the length of s1 to an int variable x.

**Int x = s1.length();**

1. Assign the first character of s1 to a char variable x.

**Char x = s1.char(0);**

1. Create a new string s3 that combines s1 with s2.

**String s3 = s1 + s2;**

1. Create a substring of s1 starting from index 1.

**String s3 = s1.substring(1);**

1. Create a substring of s1 from index 1 to index 4.

**String s3 = s1.substring(1, 5);**

1. Create a new string s3 that converts s1 to lowercase.

**String s3 = s.1toLowerCase();**

1. Create a new string s3 that converts s1 to uppercase.

string

**string s3 = s1.toUpperCase();**

1. Create a new string s3 that trims whitespaces on both ends of s1.

**String s3 = s1.trim();**

1. Assign the index of the first occurrence of the character e in s1 to an int variable x.

**Int x = s1.indexOf(‘e’);**

1. Assign the index of the last occurrence of the string abc in s1 to an int variable x.

**int x = s1.lastIndexOf(“abc”);**

6.3

Show the output of the following statements.

1. System.out.printf(“amount is %f %e\n”, 32.32, 32.32);

**amount is 32.32000 3.232000e+01**

1. System.out.printf(“amount is %5.2f%% %5.4e\n”, 32.327, 32.32);

**Amount is 32.33% 3.232000e+01**

1. System.out.printf(“%6b\n”, (1 > 2));

**false**

1. System.out.printf(“%6s\n”, “Java”);

**Java**

1. System.out.printf(“%-6b%s\n”, (1 > 2), “Java”);

**false Java**

1. System.out.printf(“%6b%-8s\n”, (1 > 2), “Java”);

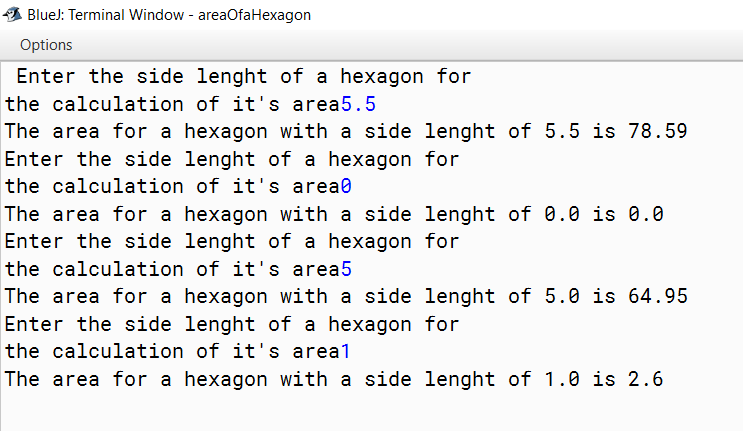
**falsejava**

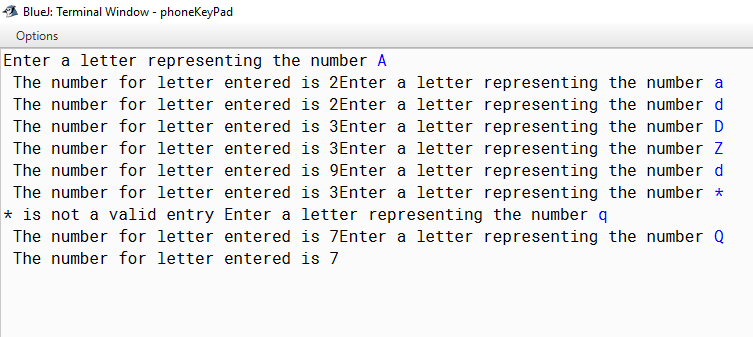
1. System.out.printf(“%,5d %,6.1f\n”, 312342, 315562.932);

**312,342 315,562.9**

1. System.out.printf(“%05d %06.1f\n”, 32, 32.32);

**00032 0032.3**

**areaOfaHexagon: RESULTS**

**phoneKeyPad: Results**

**GeneratedVehiclePlateNumbers: Results**

