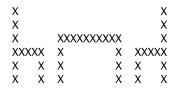
**Basics**

1. Write, compile, and test a class that displays your favorite movie quote on the screen.
2. Write, compile, and test a class that displays your favorite movie quote, the movie it comes from, the character who said it, and the year of the movie.
3. Write, compile, and test a class that displays the following pattern on the screen:



1. Carly’s Catering provides meals for parties and special events. Write a program that displays Carly’s motto, which is “Carly’s makes the food that makes it a party.”

**Data types**

1. Write a class that declares variables to hold your three initials. Display the three initials with a period following each one, as in J.M.F.
2. Meadowdale Dairy Farm sells organic brown eggs to local customers. They charge $3.25 for a dozen eggs, or 45 cents for individual eggs that are not part of a dozen. Write a class that prompts a user for the number of eggs in the order and then display the amount owed with a full explanation. For example, typical output might be, “You ordered 27 eggs. That’s 2 dozen at $3.25 per dozen and 3 loose eggs at 45 cents each for a total of $7.85.”
3. The Huntington Boys and Girls Club is conducting a fundraiser by selling chili dinners to go. The price is $7 for an adult meal and $4 for a child’s meal. Write a class that accepts the number of each type of meal ordered and display the total money collected for adult meals, children’s meals, and all meals
4. Write a class that calculates and displays the conversion of an entered number of dollars into currency denominations—20s, 10s, 5s, and 1s.
5. Write a program that accepts a number of minutes and converts it both to hours and days. For example, 6000 minutes equals 100 hours and equals 4.167 days.
6. Mad Libs is a children’s game in which they provide a few words that are then incorporated into a silly story. The game helps children understand different parts of speech because they are asked to provide specific types of words. For example,you might ask a child for a noun, another noun, an adjective, and a past-tense verb. The child might reply with such answers as table, book, silly, and studied. The newly created *Mad Lib might be:*

*Mary had a little table*

*Its book was silly as snow*

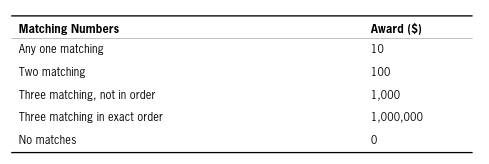
*And everywhere that Mary studied*

*The table was sure to go.*

Create a Mad Libs program that asks the user to provide at least four or five words, and then create and display a short story or nursery rhyme that uses them.

**Flow Control**

1. Write an application that asks a user to enter an integer. Display a statement that indicates whether the integer is even or odd.
2. Write an application that asks a user to enter three integers. Display them in ascending and descending order.
3. Write a program for Horizon Phones, a provider of cellular phone service. Prompt a user for maximum monthly values for talk minutes used, text messages sent, and gigabytes of data used, and then recommend the best plan for the customer’s needs. A customer who needs fewer than 500 minutes of talk and no text or data should accept Plan A at $49 per month. A customer who needs fewer than 500 minutes of talk and any text messages should accept Plan B at $55 per month. A customer who needs 500 or more minutes of talk and no data should accept either Plan C for up to 100 text messages at $61 per month or Plan D for 100 text messages or more at $70 per month. A customer who needs any data should accept Plan E for up to 2 gigabytes at $79 or Plan F for 2 gigabytes or more at $87.
4. Create a lottery game application. Generate three random numbers (see at the end info), each between 0 and 9. Allow the user to guess three numbers. Compare each of the user’s guesses to the three random numbers and display a message that includes the user’s guess, the randomly determined three-digit number, and the amount of money the user has won as follows:



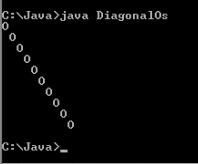
Make certain that your application accommodates repeating digits. For example, if a user guesses 1, 2, and 3, and the randomly generated digits are 1, 1, and 1, do not give the user credit for three correct guesses—just one.

1. In the game Rock Paper Scissors, two players simultaneously choose one of three options: rock, paper, or scissors. If both players choose the same option, then the result is a tie. However, if they choose differently, the winner is determined as follows:

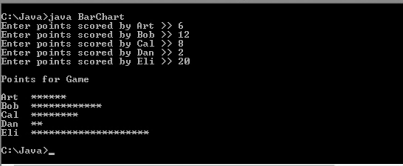
* Rock beats scissors, because a rock can break a pair of scissors.
* Scissors beats paper, because scissors can cut paper.
* Paper beats rock, because a piece of paper can cover a rock.

Create a game in which the computer randomly chooses rock, paper, or scissors. Let the user enter a number 1, 2, or 3, each representing one of the three choices.

1. Write an application that uses a loop to create the pattern of Os shown below, in which each O is displayed one additional space to the right.



1. Assume that the population of Mexico is 121 million and that the population increases 1.01 percent annually. Assume that the population of the United States is 315 million and that the population is reduced 0.15 percent annually. Write an application that displays the populations for the two countries every year until the population of Mexico exceeds that of the United States, and display the number of years it took.
2. The Huntington High School basketball team has five players named Art, Bob, Cal, Dan, and Eli. Accept the number of points scored by each player in a game and create a bar chart that illustrates the points scored, similar to the chart



1. Two people play the game of Count 21 by taking turns entering a 1, 2, or 3, which is added to a running total. The player who adds the value that makes the total exceed 21 loses the game. Create a game of Count 21 in which a player competes against the computer, and program a strategy that always allows the computer to win. On any turn, if the player enters a value other than 1, 2, or 3, force the player to reenter the value.

**Methods**

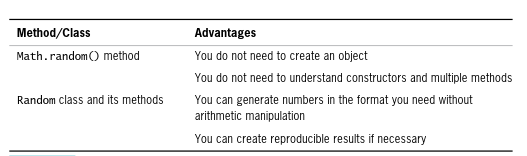
1. Create an application named NumbersDemo whose main() method holds two integer variables. Assign values to the variables. In turn, pass each value to methods named displayTwiceTheNumber(), displayNumberPlusFive(), and displayNumberSquared(). Create each method to perform the task its name implies.
2. To encourage good grades, Hermosa High School has decided to award each student a bookstore credit that is 10 times the student’s grade point average. In other words, a student with a 3.2 grade point average receives a $32 credit. Create a class that prompts a student for a name and grade point average, and then passes the values to a method that displays a descriptive message. The message uses the student’s name, echoes the grade point average, and computes and displays the credit.
3. There are 2.54 centimeters in an inch, and there are 3.7854 liters in a U.S. gallon. Create a class named MetricConversion. Its main() method accepts an integer value from a user at the keyboard, and in turn passes the entered value to two methods. One converts the value from inches to centimeters and the other converts the same value from gallons to liters. Each method displays the results with appropriate explanation.
4. Create a class named Sandwich. Data fields include a String for the main ingredient (such as “tuna”), a String for bread type (such as “wheat”), and a double for price (such as 4.99). Include methods to get and set values for each of these fields. Save the class as Sandwich.java. Create an application named TestSandwich that instantiates one Sandwich object and demonstrates the use of the set and get methods. Save this application as TestSandwich.java.
5. Create a class named Student. A Student has fields for an ID number, number of credit hours earned, and number of points earned. (For example, many schools compute grade point averages based on a scale of 4, so a three-credit-hour class in which a student earns an A is worth 12 points.) Include methods to assign values to all fields. A Student also has a field for grade point average. Include a method to compute the grade point average field by dividing points by credit hours earned. Write methods to display the values in each Student field. Save this class as Student.java. Write a class named ShowStudent that instantiates a Student object from the class you created and assign values to its fields. Compute the Student grade point average, and then display all the values associated with the Student.
6. Create a class named Lease with fields that hold an apartment tenant’s name, apartment number, monthly rent amount, and term of the lease in months. Include a constructor that initializes the name to “XXX”, the apartment number to 0, the rent to 1000, and the term to 12. Also include methods to get and set each of the fields. Include a nonstatic method named addPetFee() that adds $10 to the monthly rent value and calls a static method named explainPetPolicy() that explains the pet fee. Create a class named TestLease whose main() method declares four Lease objects. Call a getData() method three times. Within the method, prompt a user for values for each field for a Lease, and return a Lease object to the main() method where it is assigned to one of main()’s Lease objects. Do not prompt the user for values for the fourth Lease object, but let it continue to hold the default values. Then, in main(), pass one of the Lease objects to a showValues() method that displays the data. Then call the addPetFee() method using the passed Lease object and confirm that the fee explanation statement is displayed. Next, call the showValues() method for the Lease object again and confirm that the pet fee has been added to the rent. Finally, call the showValues() method with each of the other three objects; confirm that two hold the values you supplied as input and one holds the constructor default values
7. Create a class called Date that includes three instance variables—a month (type int), a day (type int) and a year (type int). Provide a constructor that initializes the three instance variables and assumes that the values provided are correct. Provide a set and a get method for each instance variable. Provide a method displayDatethat displays the month, day and year separated by forward slashes (/). Write a test application named DateTest that demonstrates class Date’s capabilities.
8. Create a class called Employee that includes three instance variables—a first name (type String), a last name (type String) and a monthly salary (double). Provide a constructor that initializes the three instance variables. Provide a set and a get method for each instance variable. If the monthly salary is not positive, do not set its value. Write a test application named EmployeeTest that demonstrates class Employee’s capabilities. Create two Employee objects and display each object’s yearly salary. Then give each Employee a 10% raise and display each Employee’s yearly salary again.

**Understanding Computer-Generated Random Numbers**

A random number is a number whose value cannot be predicted. Many types of programs use random numbers. For example, simulations that predict phenomena such as urban traffic patterns, crop production, and weather systems typically use random numbers. You might want to use random numbers to change your screen’s appearance; for example, screen savers often use random numbers to define graphics so that a changing pattern remains interesting.

Random numbers are also used in many computer game applications. When you play games with human opponents, their choices are often unpredictable (and sometimes even irrational). Computers usually are predictable and rational, so when you play a game against a computer opponent, you frequently need to generate random numbers. For example, a guessing game would not be very interesting if you were asked to guess the same number every time you played.

Most computer programming languages, including Java, come with built-in methods that generate random numbers. The random numbers are calculated based on a starting value, called a seed. The random numbers generated using these methods are not truly random; they are pseudorandom in that they produce the same set of numbers whenever the seed is the same. Therefore, if you seed a random-number generator with a constant, you always receive the same sequence of values. Many computer programs use the time of day as a random number-generating seed. For game applications, this method works well, as a player is unlikely to reset his computer’s clock and attempt to replay a game beginning at exactly the same moment in time.



**Using the Math.random() Method**

<https://www.geeksforgeeks.org/java-math-random-method-examples/>

**Using the Random Class**

<https://www.geeksforgeeks.org/java-util-random-class-java/>