* The call to **nextLine()** that follows will then properly obtain the text that the user enters
* Don’t miss the **null** argument before the values to display: this instructs Java to center the dialog on the screen rather than over another GUI element
* With && and ||, parentheses around each simple expression are optional, but there must be a set around the entire expression
* Because the while loop is a **pretest** loop, its body may never execute
* There is no Boolean expression at the top of the do-while loop (only the key word do)
* Don’t forget the semicolon after the closing parenthesis (only for do-while)
* break; // stops the loop altogether, proceeds to the next statement after the loop
* continue; /\* stops the current iteration, returns to either the Boolean expression (while, do-while) or the update statement (for) to see if another iteration is needed \*/
* while: sentinel-controlled loops, option to skip the loop body altogether
* do-while: ask-before-iterating loops, forcing at least one iteration
* for: count-controlled loops
* import java.io.\*;
* PrintWriter myOutputFile = new PrintWriter("MyFileName.txt");
* public static void main(String[] args) throws IOException
* If you’d like to add to the existing contents of a file, there’s another class, FileWriter, that can be used with PrintWriter to specify appending rather than overwriting
* FileWriter myAppend = new FileWriter("MyFileName.txt", true);
* PrintWriter myOutputFile = new PrintWriter(myAppend);
* Include the import statements for Scanner and the java.io.\* package
* To use a Scanner object with an input file, a second object of type File is required:
* File myFile = new File("MyFileName.txt");
* Scanner myInputFile = new Scanner(myFile);
* int value = myRandom.nextInt(99) + 1;
* When a method terminates, control of the program returns to the method that contains the call (not necessarily main)
* Primitive arguments are **passed by value** to a method’s parameters
* Objects are **passed by reference** to a method’s parameters of a class type
* Write method definitions inside the class brackets but outside any other method definition

import java.util.Scanner;

import java.util.Random;

public class MathTutor

{

public static void main(String[] args)

{

int number1; // A number

int number2; // Another number

int sum; // The sum of the numbers

int userAnswer; // The user's answer

Scanner keyboard = new Scanner(System.in);

Random randomNumbers = new Random();

number1 = randomNumbers.nextInt(100);

number2 = randomNumbers.nextInt(100);

System.out.println("What is the answer to the " +

"following problem?");

System.out.print(number1 + " + " +

number2 + " = ? ");

sum = number1 + number2;

userAnswer = keyboard.nextInt();

if (userAnswer == sum)

System.out.println("Correct!");

else

{

System.out.println("Sorry, wrong answer. " +

"The correct answer is " +

sum);

}

}

}

import java.util.Scanner;

import javax.swing.JOptionPane;

import java.io.\*;

public class BSindala\_Lab05

{

public static void main(String[] args) throws IOException

{

int day, highTemp, lowTemp;

File tFile = new File("MonthTemps.txt");

if (!tFile.exists())

{

JOptionPane.showMessageDialog(null, "The file " + tFile + "does not exist!");

System.exit(0);

}

Scanner inputFile = new Scanner (tFile);

PrintWriter outputFileHot = new PrintWriter("HotDays.txt");

PrintWriter outputFileCold = new PrintWriter("ColdNights.txt");

outputFileHot.println("Day \tHigh \tLow");

outputFileCold.println("Day \tHigh \tLow");

while (inputFile.hasNext())

{

day = inputFile.nextInt();

highTemp = inputFile.nextInt();

lowTemp = inputFile.nextInt();

if (highTemp >= 80)

{

outputFileHot.print(day + "\t");

outputFileHot.print(highTemp + "\t");

outputFileHot.println(lowTemp);

}

if (lowTemp <= 30)

{

outputFileCold.print(day + "\t");

outputFileCold.print(highTemp + "\t");

outputFileCold.println(lowTemp);

}

}

inputFile.close();

outputFileHot.close();

outputFileCold.close();

System.out.println("\nThe high and low temperatures have been written to their"

+ " respective files!\n");

}

}

import java.util.Scanner; // Needed for the Scanner class

public class BSindala\_Lab07

{

public static void main(String[] args)

{

char size;

int bars;

double total;

String input;

Scanner keyboard = new Scanner(System.in);

barSize();

System.out.print("Enter your selection now: ");

input = keyboard.nextLine();

size = input.charAt(0);

System.out.print("How many candy bars would you like? ");

bars = keyboard.nextInt();

total = totalAmount(size, bars); // Intializing total with the totalAmount method

System.out.printf("Your order total comes to $%.2f\n", total);

System.out.println("Please make your check payable to the school. Thanks again!");

}

public static void barSize()

{

System.out.println("Thank you for supporting our school!");

System.out.println("Please choose a candy bar size:");

System.out.println("S = Small\n" +

"M = Medium\n" +

"L = Large.");

}

public static double totalAmount(char barSize, int barQuantity)

{

double orderAmount = 0, small = 2.50, medium = 3.75, large = 5.00; // Declarations of amounts according to size

if ((barSize == 'S') || (barSize == 's'))

orderAmount = small \* barQuantity;

else if ((barSize == 'M') || (barSize == 'm'))

orderAmount = medium \* barQuantity;

else if ((barSize == 'L') || (barSize == 'l'))

orderAmount = large \* barQuantity;

return orderAmount; // Returning of the total order

}

}