**Q1.**

// Program prints all the integers in a given range using for loop

package week\_1;

public class PrintNumbers {

public static void main(String[] args) {

printNumbers(1, 10); // calling the method with 1 and 10 as arguments

}

public static void printNumbers(int from, int to) {

for (int i = from; i <= to; i++) {

System.out.print(i + " ");

}

}

}

**Q2.**

/\* program prints suggested tip amounts for a food check in a restaurant for the bill amount that user inputs \*/

package week\_1;

import static input.InputUtils.doubleInput;

public class Question\_2\_Tip\_Calculator {

public static void main(String[] args) {

double checkAmount = doubleInput("Enter the amount of the check: ");

for (int tipPercentage = 10; tipPercentage <= 30; tipPercentage += 5) {

double tipAmount = checkAmount \* tipPercentage / 100;

double totalAmount = checkAmount + tipAmount;

System.out.printf("%d%% tip $%.2f total $%.2f\n", tipPercentage, tipAmount, totalAmount);

}

}

}

**Q3.**

// Validation of college code using While Loop where code is b/w 1000 & 2999

import static input.InputUtils.intInput;

public class Question3 {

public static void main(String[] args) {

int classCode = intInput("Enter a college class code (between 1000 and 2999): ");

while (classCode < 1000 || classCode > 2999) {

System.out.println("Invalid class code. Please enter a code between 1000 and 2999.");

classCode = intInput("Enter a college class code (between 1000 and 2999): ");

}

System.out.println("Valid class code entered: " + classCode);

}

}

**Q4.**

/\*Program to create a new String from many copies of itself. The user decides how many copies to make \*/

package week\_2;

import static input.InputUtils.\*;

/\*\*

\* Finish the multiplyString method to create a String from many copies of itself.

\* The user will decide how many copies to make.

\*

\* If the method is given the String "cat" and repeats = 3 copies, it will return "catcatcat".

\* If the method is given the String "Hello" and repeats = 5 copies, it will return "HelloHelloHelloHelloHello".

\*

\* If the method is given the String "Hello" and repeats = 1 copies, it will return "Hello".

\* If the method is given the String "Hello" and repeats = 0 copies, it will return "". (An empty String)

\* If the method is given the String "Hello" and repeats = -1 copies, or any negative number it will return "". (An empty String)

\*/

public class Question\_4\_Many\_Copy\_String {

public static void main(String[] args) {

// You don't need to modify the main method.

String input = stringInput("Enter your string to multiply: ");

int copies = intInput("How many copies to make?");

String output = multiplyString(input, copies);

System.out.printf("Your String %d times is: %s%n", copies, output);

}

public static String multiplyString(String userString, int repeats) {

// return a String that is made of multiple copies of userString.

// The variable called repeats contains the number of copies.

// When userString = "Java" and repeats = 4 this method should return "JavaJavaJavaJava".

// When userString = "Java" and repeats = 2 this method should return "JavaJava".

// When userString = "Java" and repeats = 0, this method should return "" (an empty String)

// When userString = "Java" and repeats is negative, like -10, this method should return "" (an empty String)

// DON'T USE THE String repeat() LIBRARY METHOD.

// For this lab, you should solve this problem using a loop.

// Hint: Start with an empty string, then add userString to the end of that string, as many times as you need.

if (repeats <= 0) {

return "";

}

String result = "";

for (int i = 0; i < repeats; i++) {

result += userString;

}

return result;

}

}

**Q5.**

/\* program prints only the program file names from an array. i.e., any file that ends with .py or .java or .cs \*/

public class Question\_5\_FileNames {

public static void main(String[] args) {

String[] filenames = {"lab1.py", "assignment.docx", "music.mp3", "Website.cs", "Calculator.java"};

// Loop through the array of filenames

for (String filename : filenames) {

// Check if the filename ends with .py, .java or .cs

if (filename.endsWith(".py") || filename.endsWith(".java") || filename.endsWith(".cs")) {

// If it does, print the filename

System.out.println(filename);

}

}

}

}

**Q6.**

//program to remove bad data from an array of data where the connection was lost

package week\_2;

/\*\*

\*

When collecting data, it's common to get some bad data mixed in with the actual data.

For this program, imagine that you are monitoring your internet speed once an hour.

Sometimes the connection is lost completely. Due to a bug in the monitoring program,

when the connection is lost, the speed is recorded as 2,147,483,647 Mbps, (mega bits per second)

which is much too fast to be a valid speed.

1-20 Mbps is typical for an average budget home cable internet connection.

Fiber connections may be 1000Mbs or 1 gigabit.

( Question - what's special about 2,147,483,647? Why might the monitoring program record this number in particular? )

Part 1: Complete the cleanData method to remove bad data from an array of data.

A value of 2147483647 means the connection was lost, then 0 bytes were transferred.

So, replace all the 2147483647 values in the array with the number 0.

Part 2: Complete the graphData method to draw a basic vertical graph of the data.

\*/

public class Question\_6\_Clean\_And\_Graph\_Array {

public static void main(String[] args) {

// 24 speeds recorded, one per hour, over a 24-hour period

// Don't modify this array, it represents the original data provided to the program.

int[] speedsRecorded = { 8, 5, 8, 7, 8, 2147483647, 7, 8, 9, 7, 6, 6, 2147483647, 6, 2147483647, 5, 8, 7, 6, 6, 8, 9, 6, 8};

cleanData(speedsRecorded);

graphData(speedsRecorded);

}

public static void cleanData(int[] speeds) {

// Replace any element in the speeds array that has the value 2147483647, with 0.

for (int i = 0; i < speeds.length; i++) {

if (speeds[i] == 2147483647) {

speeds[i] = 0;

}

}

}

public static void graphData(int[] speeds) {

// Draw a vertical graph of the data

int maxSpeed = 0;

for (int i = 0; i < speeds.length; i++) {

if (speeds[i] > maxSpeed) {

maxSpeed = speeds[i];

}

}

for (int i = 0; i < maxSpeed; i++) {

for (int j = 0; j < speeds.length; j++) {

if (speeds[j] > i) {

System.out.print("\*");

} else {

System.out.print(" ");

}

}

System.out.println();

}

}

}

**Q7.**

/\* program to calculate the average bill for all 12 monthly bills

 user inputs values of each month's bill for last year

program adds up all the bills and calculates, and displays the average. \*/

package week\_2;

import static input.InputUtils.\*;

public class Question\_6\_Average\_Bill {

public static void main(String[] args) {

// Create an array to store the 12 monthly bills

double[] monthlyBills = new double[12];

// Use a loop to ask the user for each month's bill and store it in the array

for (int i = 0; i < monthlyBills.length; i++) {

double bill = doubleInput("Enter the bill for " + getMonthName(i) + ": ");

monthlyBills[i] = bill;

}

// Calculate the sum of all bills

double sum = 0;

for (double bill : monthlyBills) {

sum += bill;

}

// Calculate the average bill

double average = sum / monthlyBills.length;

// Print the table of months and bill amounts

System.out.println("Month\tBill");

for (int i = 0; i < monthlyBills.length; i++) {

System.out.printf("%s\t$%.2f%n", getMonthName(i), monthlyBills[i]);

}

// Print the average bill

System.out.printf("The average bill is $%.2f%n", average);

}

// An array of month names to help ask for data and display data

private static final String[] MONTHS = {"January", "February", "March", "April", "May", "June",

"July", "August", "September", "October", "November", "December"};

/\*\*

\* Returns the name of the month for the given index (0 = January, 1 = February, etc.)

\*/

private static String getMonthName(int index) {

return MONTHS[index];

}

}