EX2 (9-20)

9. public class Variable {

    public static void main(String[] args) {

        int a = 254;

        int b = 343;

        System.out.println("a + b = " + (a + b));

    }

}

10.

21.

import java.util.Scanner;

public class Input {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Hello " + sc.next());

    }

}

22.

import java.util.Scanner;

public class Input {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        String name = sc.next();

        String address = sc.next();

        System.out.println ("Name: " + name);

        System.out.println ("Address: " + address);

    }

}

23.

import java.util.Scanner;

public class Input {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int a = sc.nextInt();

        int b = sc.nextInt();

        System.out.println("a + b = " + (a + b));

    }

}

24.

import java.util.Scanner;

public class Input {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int length = sc.nextInt();

        int width = sc.nextInt();

        System.out.println("Area = " +(length \* width));

    }

}

25.

import java.util.Scanner;

public class Input {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int a = sc.nextInt();

        int b = sc.nextInt();

        System.out.println("a % b = " + (a - b \* (a / b)));

    }

}

26.

import java.util.Scanner;

public class Input {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int a = sc.nextInt();

        int b = sc.nextInt();

        System.out.println("a + b = " + (a + b));

        System.out.println("a - b = " + (a - b));

        System.out.println("a \* b = " + (a \* b));

        System.out.println("a / b = " + (a / b));

        System.out.println("a % b = " + (a % b));

    }

}

27.

import java.util.Scanner;

public class Input {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        String name = sc.next();

        int age = sc.nextInt();

        System.out.println("In 15 years, age of " + name + " will be " + (age + 15));

    }

}

28.

import java.util.Scanner;

public class Input {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int a = sc.nextInt();

        int b = sc.nextInt();

        int c = a;

        a = b;

        b = c;

        System.out.println("after swapping, a = " + a + ", b = " + b);

    }

}

29.

import java.util.Scanner;

public class Input {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        double r = sc.nextDouble();

        double pi = 3.14;

        System.out.println("Circumference = " + (2 \* pi \* r));

    }

}

30.

import java.util.Scanner;

public class Input {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        char c = (char)(sc.next().charAt(0) + 1);

        System.out.println(c);

    }

}

31

import java.util.Scanner;

public class Input {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int a = sc.nextInt();

        int b = sc.nextInt();

        System.out.println(a > b);

    }

}

32.

import java.util.Scanner;

public class Input {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();

        if (n % 2 == 0){

            System.out.println("n is an even number");

        }

        else {

        System.out.println("n is an odd number");}

    }

}

33.

import java.util.Scanner;

public class Statement {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();

        if (n == 0) {

            System.out.println("n is equal to 0");

        } else if (n < 0) {

            System.out.println("n is a negative number");

        } else {

            System.out.println("n is a positive number");

        }

}

34.

import java.util.Scanner;

public class Statement {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int a = sc.nextInt();

        int b = sc.nextInt();

        if (a >= b) {

            System.out.println("a is greater than or equal to b");

        } else {

            System.out.println("a is smaller than b");

        }

    }

}

35.

import java.util.Scanner;

public class Statement {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        String a = sc.next();

        String b = sc.next();

        if (a.equals(b)){

            System.out.println("two people have the same name");

        } else{

            System.out.println("two people don't have the same name");

        }

    }

}

36.

import java.util.Scanner;

public class Statement {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int a = sc.nextInt();

        int b = sc.nextInt();

        if (a != 0 && b != 0){

            System.out.println("a is not equal to 0 and b is not equal to 0");

        } else {

            System.out.println("a is equal to 0 or b is equal to 0");

        }

    }

}

37.

import java.util.Scanner;

public class Statement {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int a = sc.nextInt();

        int b = sc.nextInt();

        int c = sc.nextInt();

        if (a >= b && a >= c){

            System.out.println(a);

        }

        if (b >= a && b >= c){

            System.out.println(b);

        }

        if (c >= b && c >= a){

            System.out.println(c);

        }

    }

}

38.

import java.util.Scanner;

public class Statement {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int a = sc.nextInt();

        if (a >= 10 && a <= 100){

            System.out.println ( a + " is in the range [10, 100]");

        } else

        {

            System.out.println ( a + " is not in the range [10, 100]");

        }

    }

}

39.

import java.util.Scanner;

public class Statement {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int a = sc.nextInt();

        if ( a >= 0 && a <= 10){

            System.out.println ("The score is valid");

        }else{

            System.out.println ("The score is not valid");

        }

    }

}

40.

import java.util.Scanner;

public class Statement {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int a = sc.nextInt();

        int b = sc.nextInt();

        int c = sc.nextInt();

        if ( a < b && b < c && a < b )

        { System.out.println("increasing");}

        else if ( a > b && a > c && b > c ){

            System.out.println("decreasing");

        }

         else{

            System.out.println("neither increasing nor decreasing order");

        }

    }

}

41.

https://practiceit.cs.washington.edu/problem/view/bjp4/chapter2/s1-legalIntLiterals

2.1

4.3

|  |
| --- |
| public String season(int m, int d) { |
|  | if(m < 3 || (m == 3 && d <= 15) || (m == 12 && d >= 16)) { |
|  | return "Winter"; |
|  | } else if((3 < m && m < 6) || (m == 3 && d >= 16) || (m == 6 && d <= 15)) { |
|  | return "Spring"; |
|  | } else if((6 < m && m < 9) || (m == 6 && d >= 16) || (m == 9 && d <= 15)) { |
|  | return "Summer"; |
|  | } else { |
|  | return "Fall"; |
|  | } |
|  | } |

4.4

Write a method named daysInMonth that accepts a month (an integer between 1 and 12) as a parameter and returns the number of days in that month in this year. For example, the call daysInMonth(9) would return 30 because September has 30 days. Assume that the code is not being run during a leap year (that February always has 28 days).

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Month** | 1 Jan | 2 Feb | 3 Mar | 4 Apr | 5 May | 6 Jun | 7 Jul | 8 Aug | 9 Sep | 10 Oct | 11 Nov | 12 Dec |
| **Days** | 31 | 28 | 31 | 30 | 31 | 30 | 31 | 31 | 30 | 31 | 30 | 31 |

|  |
| --- |
| public int daysInMonth(int m) { |
|  | switch(m) { |
|  | case 1: |
|  | case 3: |
|  | case 5: |
|  | case 7: |
|  | case 8: |
|  | case 10: |
|  | case 12: |
|  | return 31; |
|  | case 4: |
|  | case 6: |
|  | case 9: |
|  | case 11: |
|  | return 30; |
|  | default: |
|  | return 28; |
|  | } |
|  | } |

4.8

|  |
| --- |
| /\* Write a method named smallestLargest that asks the user to enter numbers, |
|  | \* then prints the smallest and largest of all the numbers typed in by the |
|  | \* user. You may assume the user enters a valid number greater than 0 for the |
|  | \* number of numbers to read. |
|  | \*/ |
|  | public void smallestLargest() { |
|  | Scanner console = new Scanner(System.in); |
|  | System.out.print("How many numbers do you want to enter? "); |
|  | int num = console.nextInt(); |
|  | int smallest = Integer.MAX\_VALUE; |
|  | int largest = Integer.MIN\_VALUE; |
|  |  |
|  | for(int i = 1; i <= num; i++) { |
|  | System.out.print("Number " + i + ": "); |
|  | int temp = console.nextInt(); |
|  | if(temp < smallest) { |
|  | smallest = temp; |
|  | } |
|  | if(temp > largest) { |
|  | largest = temp; |
|  | } |
|  | } |
|  |  |
|  | System.out.println("Smallest = " + smallest); |
|  | System.out.println("Largest = " + largest); |
|  | } |

4.9.

|  |
| --- |
| /\* Write a method named evenSum that prompts the user for many integers and |
|  | \* print the total even sum and maximum of the even numbers. You may assume |
|  | \* that the user types at least one non-negative even integer. |
|  | \*/ |
|  | public void evenSum() { |
|  | Scanner console = new Scanner(System.in); |
|  | System.out.print("how many integers? "); |
|  | int num = console.nextInt(); |
|  | int sum = 0; |
|  | int max = Integer.MIN\_VALUE; |
|  | for(int i = 0; i < num; i++) { |
|  | System.out.print("next integer? "); |
|  | int temp = console.nextInt(); |
|  | if(temp % 2 == 0) { |
|  | sum += temp; |
|  | if(temp > max) |
|  | max = temp; |
|  | } |
|  | } |
|  | System.out.println("even sum = " + sum); |
|  | System.out.println("even max = " + max); |
|  | } |

4.10.

|  |
| --- |
| /\* Write a method named printGPA that calculates a student's grade point |
|  | \* average. The user will type a line of input containing the student's name, |
|  | \* then a number of scores, followed by that many integer scores. |
|  | \*/ |
|  | public void printGPA() { |
|  | Scanner console = new Scanner(System.in); |
|  | System.out.print("Enter a student record: "); |
|  | String name = console.next(); |
|  | int num = console.nextInt(); |
|  | double sum = 0; |
|  | for(int i = 0; i < num; i++) |
|  | sum += console.nextInt(); |
|  | System.out.println(name + "'s grade is " + sum / num); |
|  | } |

4.11

|  |
| --- |
| /\* Write a static method named longestName that reads names typed by the user |
|  | \* and prints the longest name (the name that contains the most characters) in |
|  | \* the format shown below. Your method should accept a console Scanner and an |
|  | \* integer n as parameters and should then prompt for n names. The longest |
|  | \* name should be printed with its first letter capitalized and all subsequent |
|  | \* letters in lowercase, regardless of the capitalization the user used when |
|  | \* typing in the name. If there is a tie for longest between two or more |
|  | \* names, use the tied name that was typed earliest. Also print a message |
|  | \* saying that there was a tie. |
|  | \*/ |
|  | public void longestName(Scanner console, int num) { |
|  | String longest = ""; |
|  | boolean tie = false; |
|  | for(int i = 1; i <= num; i++) { |
|  | System.out.print("name #" + i + "? " ); |
|  | String name = console.next(); |
|  | if(name.length() == longest.length()) { |
|  | tie = true; |
|  | } else if(name.length() > longest.length()) { |
|  | tie = false; |
|  | longest = name; |
|  | } |
|  | } |
|  | String capitalized = longest.substring(0, 1).toUpperCase() + |
|  | longest.substring(1).toLowerCase(); |
|  | System.out.println(capitalized + "'s name is longest"); |
|  | if(tie) |
|  | System.out.println("(There was a tie!)"); |
|  | } |

4.12.

|  |  |
| --- | --- |
|  | /\* Write a method called printTriangleType that accepts three integer |
|  | \* arguments representing the lengths of the sides of a triangle and prints |
|  | \* what type of triangle it is. The three types are equilateral, isosceles, |
|  | \* and scalene. |
|  | \*/ |
|  | public void printTriangleType(int a, int b, int c) { |
|  | if(a == b && b == c) { |
|  | System.out.println("equilateral"); |
|  | } else if(a == b || a == c || b == c) { |
|  | System.out.println("isosceles"); |
|  | } else { |
|  | System.out.println("scalene"); |
|  | } |
|  | } |

4.15.

|  |
| --- |
| /\* Write a method called printPalindrome that accepts a Scanner for the |
|  | \* console as a parameter, and prompts the user to enter one or more words and |
|  | \* prints whether the entered String is a palindrome. For an added challenge, |
|  | \* make the code case-insensitive. |
|  | \*/ |
|  | public void printPalindrome(Scanner console) { |
|  | System.out.print("Type one or more words: "); |
|  | String str = console.nextLine(); |
|  | String temp = str.toLowerCase(); |
|  | int i = 0; |
|  | int j = str.length() - 1; |
|  |  |
|  | while(i < j) { |
|  | if(temp.charAt(i) != temp.charAt(j)) { |
|  | System.out.println(str + " is not a palindrome."); |
|  | return; |
|  | } |
|  | i++; |
|  | j--; |
|  | } |
|  |  |
|  | System.out.println(str + " is a palindrome!"); |
|  | } |

4.13.

/\* Write a method called getGrade that accepts an integer representing a

\* student's grade in a course and returns that student's numerical course

\* grade. The grade can be between 0.0 (failing) and 4.0 (perfect). Assume

\* that scores are in the range of 0 to 100. For an added challenge, make your

\* method throw an IllegalArgumentException if the user passes a grade lower

\* than 0 or higher than 100.

\*/

public double getGrade(int score) {

if(score < 0 || score > 100)

throw new IllegalArgumentException();

double grade;

if(score < 60) {

grade = 0.0;

} else if(60 <= score && score <= 62) {

grade = 0.7;

} else if(63 <= score && score <= 94) {

grade = 0.8 + (score - 63) \* 0.1;

} else {

grade = 4.0;

}

return grade;

}

4.19

|  |
| --- |
| /\* Write a static method called quadrant that takes as parameters a pair of |
|  | \* real numbers representing an (x, y) point and that returns the quadrant |
|  | \* number for that point. Recall that quadrants are numbered as integers from 1 |
|  | \* to 4 with the upper-right quadrant numbered 1 and the subsequent quadrants |
|  | \* numbered in a counter-clockwise fashion. Notice that the quadrant is |
|  | \* determined by whether the x and y coordinates are positive or negative |
|  | \* numbers. If a point falls on the x-axis or the y-axis, then the method |
|  | \* should return 0. |
|  | \*/ |
|  | public int quadrant(double x, double y) { |
|  | if(x > 0 && y > 0) |
|  | return 1; |
|  |  |
|  | if(x < 0 && y > 0) |
|  | return 2; |
|  |  |
|  | if(x < 0 && y < 0) |
|  | return 3; |
|  |  |
|  | if(x > 0 && y < 0) |
|  | return 4; |
|  |  |
|  | return 0; |
|  | } |

4.20.

|  |
| --- |
| /\* Write a method named numUnique that takes three integers as parameters and |
|  | \* that returns the number of unique integers among the three. |
|  | \*/ |
|  | public int numUnique(int a, int b, int c) { |
|  | if(a == b && b == c) |
|  | return 1; |
|  |  |
|  | if(a == b || a == c || b == c) |
|  | return 2; |
|  |  |
|  | return 3; |
|  | } |

42.

import java.util.Scanner;

public class Loop {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();

        for (int i = 1; i <= n; i++){

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

        }

    }

}

43. import java.util.Scanner;

public class Loop {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int a = sc.nextInt();

        int b = sc.nextInt();

        for (int i = a; i<= b; i++){

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

        }

    }

}

44.

Input:

5

Actual output:

"5 4 3 2 1 0 -1 -2 -3 -4 -5 "

import java.util.Scanner;

public class Loop {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();

        for (int i = n; i >= -n; i--){

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

        }

    }

}

45.   
5,9

Actual output:

35

import java.util.Scanner;

public class Loop {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int a = sc.nextInt();

        int b = sc.nextInt();

        int answer = 0;

        for (int i = a; i <= b; i++){

            answer += i;

        }System.out.println(answer);

    }

}

46.

Write a program that accepts an integer n from the user and displays the sum of all odd numbers from 0 to n on the screen.

For example, if n = 7, the program will produce the following result:

16

Because 1 + 3 + 5 + 7 = 16

import java.util.Scanner;

public class Loop {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();

        int answer = 0;

        for (int i = 0; i <= n; i++){

            if (i % 2 != 0){

                answer += i;

            }

        } System.out.println(answer);

    }

}

47.

Write a program that accepts two integers a and b from the user and prints all the numbers from a to b, which are divisible by 3:

For example, if a = 1, b = 20, the program will produce the following result:

3 6 9 12 15 18

import java.util.Scanner;

public class Loop {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int a = sc.nextInt();

        int b = sc.nextInt();

        for (int i = a; i <= b; i++){

            if (i % 3 == 0) {

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

            }

        }

    }

}

48.

Write a program that accepts an integer n from the user and displays the result of n! on the screen.

For example, if n = 5, the program will display on the screen as below:

120

Because 1 \* 2 \* 3 \* 4 \* 5 = 120.

import java.util.Scanner;

public class Loop {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();

        int answer = 1;

        for (int i = 1; i <= n; i++){

            answer \*= i;

        }

        System.out.print(answer);

    }

}

49. Write a program that accepts an integer n from the user and prints the divisors of n where n > 0.

For example, if n = 12, the screen will display as below:

1 2 3 4 6 12

import java.util.Scanner;

public class Loop {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();

        for (int i = 1; i <= n; i++){

            if (n % i == 0){

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

        }

    }

}

}

50.

Write a program that prints numbers from 0 to 24 (including 0 and 24) as below:

0 1 2 3 4

5 6 7 8 9

10 11 12 13 14

15 16 17 18 19

20 21 22 23 24

import java.util.Scanner;

public class Loop {

    public static void main(String[] args) {

        for (int i = 0; i < 5; i++) {

            for (int j = 0; j < 5; j++) {

                System.out.print(i \* 5 + j + " ");

            }

            System.out.println();

        }

    }

}

51.

**Task**

Write a program that accepts an integer n from the user and prints all even numbers from n to 100 on the screen.

For example, if n = 90, the program will produce the following result:

90 92 94 96 98 100

**Theory**

You can use for loop to solve the task above. However, today we will learn how to use while loop to solve it.

While loop Syntax:

while (condition){

// If the condition is true, the code block will be executed

}

import java.util.Scanner;

public class Loop {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();

            for (int i = n; i <= 100; i++) {

            if (i % 2 == 0) {

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

            }

            }

    }

}

52.

**Task**

Write a program that accepts an integer n from the user then prints all divisors of n on the screen.

For example, if n = 12 , the program produces the following result:

6

Because all divisors of 12 are 1, 2, 3, 4, 6, 12

import java.util.Scanner;

public class Loop {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();

        int i = 1;

        int answer = 0;

        while ( i <= n){

            if(n% i == 0){

                answer++;

            }

            i++;

        }

        System.out.println(answer);

    }

}

53.

Write a program that accepts two integers a and b from the user then prints the result of ab on the screen.

For example, if a = 2, b = 3, the program will produce the following result:

8

Because 2 \* 2 \* 2 = 8.

import java.util.Scanner;

public class Loop {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int a = sc.nextInt();

        int b = sc.nextInt();

        int answer = 1;

        for (; b > 0; b--) {

            answer \*= a;

        }

            System.out.print(answer);

    }

}

54.

Write a program that accepts two integers a and b from the user then prints all numbers from a to b, which are divisible by 3 and 5.

For example, if a = 1, b = 50, the program produces the following result:

15 30 45

import java.util.Scanner;

public class Loop {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int a = sc.nextInt();

        int b = sc.nextInt();

        while (a <= b){

            if( a % 3 == 0 && a % 5 == 0){

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

            }

            a++;

        }

    }

}

55

Fill in the blank (...) to complete the program that prints all numbers from 1 to 50 on the screen.

public class Loop {

    public static void main(String[] args) {

        for (int i = 1; i <= 100; i++) {

            if (i == 51){

                break;

            }

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

        }

    }

}

56.

2.1

|  |  |
| --- | --- |
|  | /\* In physics, a common useful equation for finding the position s of a body |
|  | \* in linear motion at a given time t, based on its initial position s0, |
|  | \* initial velocity v0, and rate of acceleration a, is the following: |
|  | \* |
|  | \* s = s0 + v0 t + ½ at2 |
|  | \* |
|  | \* Write code to declare variables for s0 with a value of 12.0, v0 with a |
|  | \* value of 3.5, a with a value of 9.8, and t with a value of 10, and then |
|  | \* write the code to compute s on the basis of these values. At the end of |
|  | \* your code, print the value of your variable s to the console. |
|  | \*/ |
|  | double s0 = 12.0; |
|  | double v0 = 3.5; |
|  | double a = 9.8; |
|  | double t = 10.0; |
|  |  |
|  | double s = s0 + v0 \* t + 0.5 \* a \* t \* t; |
|  | System.out.println(s); |

2.2

|  |  |
| --- | --- |
|  | /\* Write a for loop that produces the following output: |
|  | \* 1 4 9 16 25 36 49 64 81 100 |
|  | \* For added challenge, try to modify your code so that it does not need to |
|  | \* use the \* multiplication operator. |
|  | \*/ |
|  | for(int i = 1, inc = 3; i <= 100; inc += 2) { |
|  | System.out.print(i + " "); |
|  | i += inc; |
|  | } |

2.3

|  |
| --- |
| /\* Write a piece of code that uses a for loop to compute and print the first |
|  | \* 12 Fibonacci numbers. |
|  | \*/ |
|  | int f0 = 1; |
|  | int f1 = 1; |
|  |  |
|  | System.out.print(f0 + " "); |
|  | System.out.print(f1 + " "); |
|  |  |
|  | for(int fk = f1 + f0; fk <= 144; fk = f1 + f0) { |
|  | System.out.print(fk + " "); |
|  | f0 = f1; |
|  | f1 = fk; |
|  | } |

2.4

|  |
| --- |
| /\* Write for loops to produce the following output: |
|  | \* |
|  | \* \*\*\*\*\* |
|  | \* \*\*\*\*\* |
|  | \* \*\*\*\*\* |
|  | \* \*\*\*\*\* |
|  | \*/ |
|  | for(int i = 0; i < 4; i++) { |
|  | for(int j = 0; j < 5; j++) |
|  | System.out.print("\*"); |
|  | System.out.println(); |
|  | } |

2.5

|  |
| --- |
| /\* Write for loops to produce the following output: |
|  | \* |
|  | \* \* |
|  | \* \*\* |
|  | \* \*\*\* |
|  | \* \*\*\*\* |
|  | \* \*\*\*\*\* |
|  | \*/ |
|  | for(int i = 1; i <= 5; i++) { |
|  | for(int j = 0; j < i; j++) |
|  | System.out.print("\*"); |
|  | System.out.println(); |
|  | } |

2.6

|  |  |
| --- | --- |
|  | /\* Write for loops to produce the following output: |
|  | \* 1 |
|  | \* 22 |
|  | \* 333 |
|  | \* 4444 |
|  | \* 55555 |
|  | \* 666666 |
|  | \* 7777777 |
|  | \*/ |
|  | for(int i = 1; i < 8; i++) { |
|  | for(int j = 0; j < i; j++) |
|  | System.out.print(i); |
|  | System.out.println(); |
|  | } |

2.7

|  |
| --- |
| /\* Write nested for loops to produce the following output: |
|  | \* |
|  | \* 1 |
|  | \* 2 |
|  | \* 3 |
|  | \* 4 |
|  | \* 5 |
|  | \*/ |
|  | for(int i = 1; i <= 5; i++) { |
|  | for(int j = 5 - i; j > 0; j--) { |
|  | System.out.print(" "); |
|  | } |
|  | System.out.println(i); |
|  | } |

2.8

|  |
| --- |
| /\* Write nested for loops to produce the following output: |
|  | \* |
|  | \* 1 |
|  | \* 22 |
|  | \* 333 |
|  | \* 4444 |
|  | \* 55555 |
|  | \*/ |
|  | for(int i = 1; i <= 5; i++) { |
|  | for(int j = 5 - i; j > 0; j--) |
|  | System.out.print(" "); |
|  |  |
|  | for(int j = 0; j < i; j++) |
|  | System.out.print(i); |
|  |  |
|  | System.out.println(); |
|  | } |

2.9.

|  |
| --- |
| /\* Write for loops to produce the following output, with each line 40 |
|  | \* characters wide: |
|  | \* |
|  | \* ---------------------------------------- |
|  | \* \_-^-\_-^-\_-^-\_-^-\_-^-\_-^-\_-^-\_-^-\_-^-\_-^- |
|  | \* 1122334455667788990011223344556677889900 |
|  | \* ---------------------------------------- |
|  | \*/ |
|  | for(int i = 0; i < 40; i++) |
|  | System.out.print("-"); |
|  |  |
|  | System.out.println(); |
|  |  |
|  | for(int i = 0; i < 10; i++) |
|  | System.out.print("\_-^-"); |
|  |  |
|  | System.out.println(); |
|  |  |
|  | for(int j = 0; j < 2; j++) { |
|  | for(int i = 1; i <= 10; i++) { |
|  | System.out.print(i%10); |
|  | System.out.print(i%10); |
|  | } |
|  | } |
|  |  |
|  | System.out.println(); |
|  |  |
|  | for(int i = 0; i < 40; i++) |
|  | System.out.print("-"); |

2.10.

|  |
| --- |
| /\* Modify your code from the previous exercise so that it could easily be |
|  | \* modified to display a different range of numbers (instead of 1234567890) |
|  | \* and a different number of repetitions of those numbers (instead of 60 total |
|  | \* characters), with the vertical bars still matching up correctly. Write a |
|  | \* complete class named NumbersOutput. Use two class constants instead of |
|  | \* "magic numbers,", with one constant set to 6 for the number of repetitions, |
|  | \* and the other set to 10 for the range of numbers. Put the for loop code in |
|  | \* your class's main method. |
|  | \*/ |
|  | public class NumbersOutput { |
|  | public static final int NUM\_REPETITIONS = 6; |
|  | public static final int NUM\_RANGE = 10; |
|  |  |
|  | public static void main(String[] args) { |
|  | for(int i = 0; i < NUM\_REPETITIONS; i++) { |
|  | for(int j = 0; j < NUM\_RANGE - 1; j++) |
|  | System.out.print(" "); |
|  |  |
|  | System.out.print("|"); |
|  | } |
|  |  |
|  | System.out.println(); |
|  |  |
|  | for(int i = 1; i <= NUM\_REPETITIONS \* NUM\_RANGE; i++) |
|  | System.out.print(i % NUM\_RANGE); |
|  | } |
|  | } |

2.10.

|  |
| --- |
| for(int i = 0; i < 6; i++) { |
|  | for(int j = 0; j < 9; j++) |
|  | System.out.print(" "); |
|  | System.out.print("|"); |
|  | } |
|  |  |
|  | System.out.println(); |
|  |  |
|  | for(int i = 1; i <= 60; i++) |
|  | System.out.print(i%10); |

|  |
| --- |
| /\* Write a Java program called DollarFigure that produces the following |
|  | \* output. Use nested for loops to capture the structure of the figure. |
|  | \* $$$$$$$\*\*\*\*\*\*\*\*\*\*\*\*\*\*$$$$$$$ |
|  | \* \*\*$$$$$$\*\*\*\*\*\*\*\*\*\*\*\*$$$$$$\*\* |
|  | \* \*\*\*\*$$$$$\*\*\*\*\*\*\*\*\*\*$$$$$\*\*\*\* |
|  | \* \*\*\*\*\*\*$$$$\*\*\*\*\*\*\*\*$$$$\*\*\*\*\*\* |
|  | \* \*\*\*\*\*\*\*\*$$$\*\*\*\*\*\*$$$\*\*\*\*\*\*\*\* |
|  | \* \*\*\*\*\*\*\*\*\*\*$$\*\*\*\*$$\*\*\*\*\*\*\*\*\*\* |
|  | \* \*\*\*\*\*\*\*\*\*\*\*\*$\*\*$\*\*\*\*\*\*\*\*\*\*\*\* |
|  | \*/ |
|  | public class DollarFigure { |
|  | public static void main(String[] args) { |
|  | for(int i = 1; i <= 7; i++) { |
|  | for(int j = 0; j < 2 \* i - 2; j++) |
|  | System.out.print("\*"); |
|  |  |
|  | for(int j = 0; j < -i + 8; j++) |
|  | System.out.print("$"); |
|  |  |
|  | for(int j = 0; j < -2 \* i + 16; j++) |
|  | System.out.print("\*"); |
|  |  |
|  | for(int j = 0; j < -i + 8; j++) |
|  | System.out.print("$"); |
|  |  |
|  | for(int j = 0; j < 2 \* i - 2; j++) |
|  | System.out.print("\*"); |
|  |  |
|  | System.out.println(); |
|  | } |
|  | } |
|  | } |

DOLLA FIGURE 2

|  |
| --- |
| /\* |
|  | \* Modify your DollarFigure program to become a new program called |
|  | \* DollarFigure2 that uses a global constant for the figure's height. |
|  | \* |
|  | \* Size 3: |
|  | \* $$$\*\*\*\*\*\*$$$ |
|  | \* \*\*$$\*\*\*\*$$\*\* |
|  | \* \*\*\*\*$\*\*$\*\*\*\* |
|  | \* |
|  | \* Size 5: |
|  | \* $$$$$\*\*\*\*\*\*\*\*\*\*$$$$$ |
|  | \* \*\*$$$$\*\*\*\*\*\*\*\*$$$$\*\* |
|  | \* \*\*\*\*$$$\*\*\*\*\*\*$$$\*\*\*\* |
|  | \* \*\*\*\*\*\*$$\*\*\*\*$$\*\*\*\*\*\* |
|  | \* \*\*\*\*\*\*\*\*$\*\*$\*\*\*\*\*\*\*\* |
|  | \*/ |
|  |  |
|  | public class DollarFigure2 { |
|  | public static final int size = 5; |
|  | public static void main(String[] args) { |
|  | for (int i = 0; i < size; i++) { |
|  |  |
|  | for (int j = 1; j <= 2 \* i; j++) { |
|  | System.out.print("\*"); |
|  | } |
|  |  |
|  | for (int j = 1; j <= size - i; j++) { |
|  | System.out.print("$"); |
|  | } |
|  |  |
|  | for (int j = 1; j <= 2 \* size - 2 \* i; j++) { |
|  | System.out.print("\*"); |
|  | } |
|  |  |
|  | for (int j = 1; j <= size - i; j++) { |
|  | System.out.print("$"); |
|  | } |
|  |  |
|  | for (int j = 1; j <= 2 \* i; j++) { |
|  | System.out.print("\*"); |
|  | } |
|  |  |
|  | System.out.println(); |
|  | } |
|  | } |
|  | } |

|  |  |
| --- | --- |
|  | /\*Modify your StarFigure code from the previous problem to use a constant for the size. The outputs below use a constant size of 3 (left) and 6 (right): |
|  |  |
|  | size 3 |
|  | ////////\\\\\\\\ |
|  | ////\*\*\*\*\*\*\*\*\\\\ |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |
|  | size 6 |
|  | ////////////////////\\\\\\\\\\\\\\\\\\\\ |
|  | ////////////////\*\*\*\*\*\*\*\*\\\\\\\\\\\\\\\\ |
|  | ////////////\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\\\\\\\\\\\\ |
|  | ////////\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\\\\\\\\ |
|  | ////\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\\\\ |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |
|  | (You must solve this problem using only ONE public static final constant, not multiple constants; and its value must be used |
|  | in the way described in this problem.)\*/ |
|  |  |
|  | public class StarFigure2 { |
|  | public static final int R = 7; |
|  |  |
|  | public static void main(String[] args) { |
|  | for (int i = 1; i <= R; i++) { |
|  | for (int j = 1; j <= (4\*R) - 4 \* i; j++) { |
|  | System.out.print("/"); |
|  |  |
|  | } |
|  | for (int j = 1; j <= 8 \* i - 8; j++) { |
|  | System.out.print("\*"); |
|  | } |
|  | for (int j = 1; j <= (4\*R) - 4 \* i; j++) { |
|  | System.out.print("\\"); |
|  | } |
|  | System.out.println(); |
|  | } |
|  | } |
|  | } |

2.20

|  |  |
| --- | --- |
|  | /\*Write a program in a class named StarFigure that produces the following output using for loops. |
|  |  |
|  | ////////////////\\\\\\\\\\\\\\\\ |
|  | ////////////\*\*\*\*\*\*\*\*\\\\\\\\\\\\ |
|  | ////////\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\\\\\\\\ |
|  | ////\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\\\\ |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |
|  | \*/ |
|  | public class StarFigure { |
|  |  |
|  | public static void main(String[] args) { |
|  | for (int i = 1; i <= 5; i++) { |
|  | for (int j = 1; j <= 20 - 4 \* i; j++) { |
|  | System.out.print("/"); |
|  |  |
|  | } |
|  | for (int j = 1; j <= 8 \* i - 8; j++) { |
|  | System.out.print("\*"); |
|  | } |
|  | for (int j = 1; j <= 20 - 4 \* i; j++) { |
|  | System.out.print("\\"); |
|  | } |
|  | System.out.println(); |
|  | } |
|  | } |
|  | }  2.19.   |  | | --- | | /\*Write a Java program in a class named Window that produces the preceding figure as output. Use nested for loops to print the repeated parts of the figure. | |  | Once you get it to work, add one class constant to your program so that the size of the figure can be changed simply by | |  | changing that constant's value. The example output shown is at a constant size of 3, but if you change the constant, the figure | |  | should grow larger and wider proportionally. | |  |  | |  | +===+===+ | |  | | | | | |  | | | | | |  | | | | | |  | +===+===+ | |  | | | | | |  | | | | | |  | | | | | |  | +===+===+ | |  |  | |  | (You must solve this problem using only ONE public static final constant, not multiple constants; and its value must be used | |  | in the way described in this problem.)\*/ | |  | public class Window { | |  | public static final int R = 7; | |  |  | |  | public static void main(String[] args) { | |  |  | |  | System.out.print("+"); | |  | for(int i = 1; i <= 2; i++) { | |  | for(int j = 1; j <= R; j++) | |  | System.out.print("="); | |  | System.out.print("+"); | |  | } | |  | System.out.println(); | |  | for(int k = 1; k <= R; k++) { | |  | for(int i = 1; i <= 3; i++) { | |  | System.out.print("|"); | |  | for(int j = 1; j <= R; j++) | |  | System.out.print(" "); | |  | } | |  | System.out.println(); | |  | } | |  | System.out.print("+"); | |  | for(int i = 1; i <= 2; i++) { | |  | for(int j = 1; j <= R; j++) | |  | System.out.print("="); | |  | System.out.print("+"); | |  | } | |  | window(); | |  | } | |  | public static void window() { | |  | System.out.println(); | |  | for(int k = 1; k <= R; k++) { | |  | for(int i = 1; i <= 3; i++) { | |  | System.out.print("|"); | |  | for(int j = 1; j <= R; j++) | |  | System.out.print(" "); | |  | } | |  | System.out.println(); | |  | } | |  | System.out.print("+"); | |  | for(int i = 1; i <= 2; i++) { | |  | for(int j = 1; j <= R; j++) | |  | System.out.print("="); | |  | System.out.print("+"); | |  | } | |  | } | |  | } | |

2.17.

|  |  |
| --- | --- |
|  | /\*Modify the SlashFigure program from the previous exercise to produce a new program SlashFigure2 that uses a global constant |
|  | for the figure's height. The previous output used a constant height of 6. Here is the outputs for a constant height of 4 and |
|  | 7 respectively: (If you previously solved Self-Check problems 34 and 35 in the book, you will have created a loop table that |
|  | will be useful in solving this problem. Use it!) |
|  |  |
|  | size 4 |
|  | !!!!!!!!!!!!!! |
|  | \\!!!!!!!!!!// |
|  | \\\\!!!!!!//// |
|  | \\\\\\!!////// |
|  |  |
|  | size 7 |
|  | !!!!!!!!!!!!!!!!!!!!!!!!!! |
|  | \\!!!!!!!!!!!!!!!!!!!!!!// |
|  | \\\\!!!!!!!!!!!!!!!!!!//// |
|  | \\\\\\!!!!!!!!!!!!!!////// |
|  | \\\\\\\\!!!!!!!!!!//////// |
|  | \\\\\\\\\\!!!!!!////////// |
|  | \\\\\\\\\\\\!!//////////// |
|  |  |
|  | (You must solve this problem using only ONE public static final constant, not multiple constants; and its value must be used |
|  | in the way described in this problem.)\*/ |
|  |  |
|  | public class SlashFigure2 { |
|  | public static final int r = 4; |
|  |  |
|  | public static void main(String[] args) { |
|  | for(int i = 1; i <= r; i++) { |
|  | for(int j = 0; j < 2 \* i - 2; j++) |
|  | System.out.print("\\"); |
|  | for(int j = 0; j < -4 \* i + 4 \* r + 2; j++) |
|  | System.out.print("!"); |
|  | for(int j = 0; j < 2 \* i - 2; j++) |
|  | System.out.print("/"); |
|  | System.out.println(); |
|  |  |
|  | } |
|  | } |
|  | } |

2.16.

|  |
| --- |
| /\*Write a Java program in a class named SlashFigure that produces the following output. Use nested for loops to capture the structure of the figure. (If you previously solved Self-Check problems 34 and 35 in the book, you will have created a loop table that will be useful in solving this problem. Use it!) |
|  |  |
|  | !!!!!!!!!!!!!!!!!!!!!! |
|  | \\!!!!!!!!!!!!!!!!!!// |
|  | \\\\!!!!!!!!!!!!!!//// |
|  | \\\\\\!!!!!!!!!!////// |
|  | \\\\\\\\!!!!!!//////// |
|  | \\\\\\\\\\!!////////// |
|  |  |
|  | \*/ |
|  |  |
|  | public class SlashFigure { |
|  |  |
|  | public static void main(String[] args) { |
|  | for(int i = 1; i <= 6; i++) { |
|  | for(int j = 0; j < 2 \* i - 2; j++) |
|  | System.out.print("\\"); |
|  | for(int j = 0; j < -4 \* i + 26; j++) |
|  | System.out.print("!"); |
|  | for(int j = 0; j < 2 \* i - 2; j++) |
|  | System.out.print("/"); |
|  | System.out.println(); |
|  |  |
|  | } |
|  | } |
|  | } |

2.15.

|  |
| --- |
| /\*Write a method called printDesign that produces the following output. Use nested for loops to capture the structure of the figure. |
|  |  |
|  | -----1----- |
|  | ----333---- |
|  | ---55555--- |
|  | --7777777-- |
|  | -999999999- |
|  |  |
|  | \*/ |
|  | public static void printDesign() { |
|  | for(int i = 1; i <= 9; i+=2) { |
|  | for(int j = 0; j < (11 - i) / 2; j++) |
|  | System.out.print("-"); |
|  |  |
|  | for(int j = 0; j < i; j++) |
|  | System.out.print(i); |
|  |  |
|  | for(int j = 0; j < (11 - i) / 2; j++) |
|  | System.out.print("-"); |
|  |  |
|  | System.out.println(); |
|  | } |
|  | } |

2.14.

|  |
| --- |
| /\*Modify your code from the previous problem to produce the following output: |
|  |  |
|  |  |
|  | 999999999888888887777777666666555554444333221 |
|  | 999999999888888887777777666666555554444333221 |
|  | 999999999888888887777777666666555554444333221 |
|  | 999999999888888887777777666666555554444333221 |
|  |  |
|  | \*/ |
|  |  |
|  | for(int i = 1; i <= 4; i++) { |
|  | for(int j = 9; j >= 1; j--) { |
|  | for(int k = j - 1; k >= 0; k--) |
|  | System.out.print(j); |
|  | } |
|  | System.out.println(); |
|  | } |

2.13.

|  |
| --- |
| /\*Modify your code from the previous problem to produce the following output: |
|  |  |
|  |  |
|  | 99999888887777766666555554444433333222221111100000 |
|  | 99999888887777766666555554444433333222221111100000 |
|  | 99999888887777766666555554444433333222221111100000 |
|  | 99999888887777766666555554444433333222221111100000 |
|  | 99999888887777766666555554444433333222221111100000 |
|  |  |
|  | \*/ |
|  | for(int i = 1; i <= 5; i++) { |
|  | for(int j = 9; j >= 0; j--) { |
|  | for(int k = 1; k <= 5; k++) |
|  | System.out.print(j); |
|  | } |
|  | System.out.println(); |
|  | } |

2.11

|  |
| --- |
| /\*Write nested for loops that produce the following output: |
|  |  |
|  |  |
|  | 000111222333444555666777888999 |
|  | 000111222333444555666777888999 |
|  | 000111222333444555666777888999 |
|  |  |
|  | \*/ |
|  |  |
|  | for(int i = 1; i <= 3; i++) { |
|  | for(int j = 0; j <= 9; j++) { |
|  | for(int k = 1; k <= 3; k++) |
|  | System.out.print(j); |
|  | } |
|  | System.out.println(); |
|  | } |

2.\\

|  |
| --- |
| public class NumbersOutput { |
|  | public static final int repetitions = 6; |
|  | public static final int range = 10; |
|  |  |
|  | public static void main(String[] args) { |
|  | for(int k = 1; k <= repetitions; k++) { |
|  | for(int i = 1; i < range; i++) |
|  | System.out.print(" "); |
|  | System.out.print("|"); |
|  | } |
|  | System.out.println(); |
|  | for(int j = 1; j <= (range \* repetitions); j++) |
|  | System.out.print(j%range); |
|  | } |
|  | } |

public static void game() {  
 Random rd = new Random();  
 int answer = rd.nextInt(101);  
 System.*out*.println("Tôi đang nghĩ tới một số may mắn từ 1 tới 100");  
  
 int soMayMan;  
 int soLanDoan = 0;  
 Scanner input = new Scanner(System.*in*);  
 do {  
 soLanDoan++;  
 System.*out*.println("Bạn đoán số nào? ");  
 soMayMan = input.nextInt();  
 if (soMayMan == answer) {  
 System.*out*.println("Chúc mừng bạn đã đoán đúng con số may mắn sau " + soLanDoan + "lần dự đoán.");  
 break;  
 }  
 if (soMayMan > answer) {  
 System.*out*.println("Số may mắn nhỏ hơn số dự đoán.");  
 } else {  
 System.*out*.println("Số may mắn lớn hơn số dự đoán.");  
 }  
 while (soLanDoan < 7) ;  
 if (soLanDoan == 7 && soMayMan != answer) {  
 System.*out*.println("Bạn đã đoán quá 7 lần");  
 }  
 //input.close();  
  
 public static void (String[] args){  
 Sanner sc = new Scanner (System.*in*);  
 do{  
 *game*();  
 Sc=new Scanner (system.in);  
 System.*out*.println("Bạn có muốn chơi lại không? (Y/N)");  
 String check = sc.nextLine ();  
 if (check.equalsIgnoreCase("No") || check.equalsIgnoreCase("No")|)  
 break;  
 }  
 while (true);  
 System.*out*.println("Thanks for using!");  
 sc.close();  
 }  
 }

|  |
| --- |
|  |
| //Homework 5 |
|  |  |
|  | import java.util.Scanner; |
|  | import java.util.Random; |
|  |  |
|  | public class guess { |
|  |  |
|  | //method to introduces the game to the user |
|  | public static void intro () { |
|  | System.out.println(""); |
|  |  |
|  | } |
|  |  |
|  | //method to play one game with the user (just one game, not multiple games) |
|  | **public static int oneGame () {** |
|  | Random rand = new Random(); |
|  | Scanner input = new Scanner(System.in); |
|  |  |
|  | int guess; //user's guess |
|  | int actual = rand.nextInt(100) + 1; //random number between 1=100 for actual number |
|  | int tries = 1; //number of tries user takes |
|  |  |
|  |  |
|  | System.out.println(""); |
|  | //System.out.println("Actual: " + actual); |
|  | System.out.println("I'm thinking of a number between 1 and 100..."); |
|  | System.out.print("Your guess? "); |
|  | guess = input.nextInt(); |
|  |  |
|  | while(guess != actual) { |
|  | if(guess > actual) { |
|  | System.out.println("It\'s lower."); |
|  | } else { |
|  | System.out.println("It\'s higher."); |
|  | } |
|  |  |
|  | tries++; |
|  |  |
|  | System.out.print("Your guess? "); |
|  | guess = input.nextInt(); |
|  | } |
|  |  |
|  | System.out.println("You got it right in " + tries + " guesses!"); |
|  |  |
|  | return tries; |
|  | } |
|  |  |
|  | //a method to report overall results to the user |
|  | public static void results (int games, int guesses, int best) { |
|  |  |
|  | System.out.println(""); |
|  | System.out.println("Overall results:"); |
|  | System.out.println(" total games = " + games); |
|  | System.out.println(" total guesses = " + guesses); |
|  | System.out.println(" guesses/game = " + (guesses/games)); |
|  | System.out.println(" best game = " + best); |
|  |  |
|  | } |
|  |  |
|  | public static void main (String[] args) { |
|  | Scanner input = new Scanner(System.in); |
|  |  |
|  | int games = 1; //number of games played |
|  | int guesses = 0; //number of guesses in total; |
|  | int best\_game = 0; //number of tries on the best game |
|  | String play\_again; //keep playing |
|  | int tries; //current game's number of tries |
|  |  |
|  | intro(); |
|  |  |
|  | best\_game = oneGame(); |
|  | guesses = best\_game; |
|  | System.out.print("Do you want to play again? "); |
|  | play\_again = input.next(); |
|  |  |
|  | //check if play\_again is inputted correctly |
|  | if (!(play\_again.substring(0,1).equalsIgnoreCase("n")) && !(play\_again.substring(0,1).equalsIgnoreCase("y"))) { |
|  | System.out.println("Sorry... I didn't get that"); |
|  | System.out.print("Do you want to play again? "); |
|  | play\_again = input.next(); |
|  | } |
|  |  |
|  | while(play\_again.substring(0,1).equalsIgnoreCase("y")) { |
|  | tries = oneGame(); |
|  | guesses = guesses + tries; |
|  |  |
|  | //set best game to be the current game number of tries |
|  | if (tries < best\_game){ |
|  | best\_game = tries; |
|  | } |
|  |  |
|  | games++; |
|  | } |
|  |  |
|  | results(games, guesses, best\_game); |
|  |  |
|  | } |
|  |  |
|  |  |
|  | } |
|  |  |

System.*out*.println("You got it right in " + tries + " guesses!");  
  
 return tries;  
 }  
  
 for (; ; ) {  
 while (true) {  
 if (totalGuess < 7) {  
 System.*out*.print("Your guess? ");  
 int Guessnumber = sc.nextInt();  
 totalGuess++;  
 if (Guessnumber == answer) {  
 System.*out*.println("You got it right in " + totalGuess + " guesses");  
  
 } else {  
 if (Guessnumber < answer) {  
 System.*out*.println("It's higher");  
 } else {  
 System.*out*.println("It's lower");  
 }  
 }  
 }  
 if (totalGuess >= 7) {  
 do {  
 String re1 = "y";  
 String re2 = "yes";  
 Scanner input = new Scanner(System.*in*);  
 System.*out*.println("Do you want to play again? ");  
 String rep = new Scanner(System.*in*).nextLine();  
 if (rep.equalsIgnoreCase(re1) || rep.equalsIgnoreCase(re2)) {  
  
 } else {  
 System.*out*.println("End game");  
 return;  
 }  
  
  
 } while (false);  
  
  
 }  
 }  
 }  
  
}

public static void ask() {  
 String rep1;  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.print("Do you want to play again? ");  
 rep1 = sc.next();  
 if ((rep1.equalsIgnoreCase("y")) || (rep1.equalsIgnoreCase("yes"))) {  
 *playGame*();  
 } else {  
 System.*out*.println("Endgame");  
 *playGameend*();  
 }  
 }  
 public static int playGameend() {  
  
 }  
 public static int playGame () {  
 Random rand = new Random();  
 Scanner input = new Scanner(System.*in*);  
  
 int guess; //số người chơi đoán  
 int answer = rand.nextInt(101); //đáp án của chương trình random 0-100  
 int countguess = 1; //số lần đoán  
 int totalguess = 1;  
 int totalgame =1;  
 double gameavg;  
  
 System.*out*.print("Your guess? ");  
 guess = input.nextInt();  
  
 while(guess != answer) {  
 if(guess > answer) {  
 System.*out*.println("It\'s lower.");  
 } else {  
 System.*out*.println("It\'s higher.");  
 }  
  
 countguess++;  
  
  
 System.*out*.print("Your guess? ");  
 guess = input.nextInt();  
 }  
  
  
 System.*out*.println("You got it right in " + countguess + " guesses!");  
  
 *ask*();  
 return countguess;  
 }  
  
  
  
 }  
  
  
  
  
 public static void main (String[] args) {  
 Scanner input = new Scanner(System.*in*);  
  
  
 int games = 1; //number of games played  
 int guesses = 0; //number of guesses in total;  
 int best\_game = 0; //number of tries on the best game  
 String play\_again; //keep playing  
 int tries; //current game's number of tries  
  
  
 *intro*();  
  
  
 best\_game = *playGame*();  
 guesses = best\_game;  
 *ask*();  
  
  
 while(play\_again.substring(0,1).equalsIgnoreCase("y")) {  
 tries = *playGame*();  
 guesses = guesses + tries;  
  
  
 //set best game to be the current game number of tries  
 if (tries < best\_game){  
 best\_game = tries;  
 }  
  
  
 games++;  
 }  
  
  
 *results*();  
  
  
 }  
 public static void results () {  
 int games;  
 int guesses;  
 int best;  
  
  
 System.*out*.println("");  
 System.*out*.println("Overall results:");  
 System.*out*.println(" total games = " + games);  
 System.*out*.println(" total guesses = " + guesses);  
 System.*out*.println(" guesses/game = " + (guesses/games));  
 System.*out*.println(" best game = " + best);  
  
  
 }  
  
  
  
}

import java.util.Scanner;  
import java.util.Random;  
  
  
public class LuckyNumber {  
 int totalGame = 0;  
  
 public static void intro() {  
 System.*out*.println("Hello!");  
 System.*out*.println("Let's play a game!");  
 System.*out*.println("I'm thinking of a number between 0 and 100. What is it?");  
 }  
 // mỗi lượt chơi:  
 public static int oneGame () {  
 Random rand = new Random();  
 Scanner input = new Scanner(System.*in*);  
  
 int guess; //nguoi choi doan  
 int answer = rand.nextInt(101); //dap so ngau nhien  
 int countGuess = 1; //so lan doan  
 int totalGuess =1; //cả chương trình  
  
 System.*out*.print("Your guess? ");  
 guess = input.nextInt();  
  
  
 while(guess != answer) {  
 if(guess > answer) {  
 System.*out*.println("It\'s lower.");  
 } else {  
 System.*out*.println("It\'s higher.");  
 }  
  
 countGuess++;  
 totalGuess++;  
  
 System.*out*.print("Your guess? ");  
 guess = input.nextInt();  
 }  
  
  
 System.*out*.println("You got it right in " + countGuess + " guesses!");  
  
  
 return countGuess;  
 }  
  
  
 //báo cáo kết quả  
 public static void results (int totalGame, int totalGuess, int best, double avg) {  
  
 System.*out*.println("");  
 System.*out*.println("Overall results:");  
 System.*out*.println(" total games = " + totalGame);  
 System.*out*.println(" total guesses = " + totalGuess);  
 System.*out*.println(" guesses/game = " + avg);  
 System.*out*.println(" best game = " + best);  
  
 }  
 // chơi game  
 public static void main (String[] args) {  
 Scanner input = new Scanner(System.*in*);  
  
  
 int totalGame = 1; //number of games played  
 int totalGuess = 0; //tổng số lượt đoán  
 int best\_game = 0; //number of tries on the best game  
 String rep1; //chơi tiếp hay không  
 int countGuess= 0;  
  
  
 double avg = (double) totalGuess/totalGame;  
 *intro*();  
  
  
 best\_game = *oneGame*();  
 countGuess = best\_game;  
 System.*out*.print("Do you want to play again? ");  
 rep1 = input.next();  
  
  
 //check if play\_again is inputted correctly  
 if ((rep1.equalsIgnoreCase("y"))||(rep1.equalsIgnoreCase("yes"))) {  
 countGuess = *oneGame*();  
 totalGuess = totalGuess + countGuess;  
  
  
 //set best game to be the current game number of tries  
 if (countGuess < best\_game){  
 best\_game = countGuess;  
 }  
  
  
 totalGame++;  
 }  
  
  
 *results*(totalGame, totalGuess, best\_game, avg);  
  
  
 }  
  
  
  
  
}

//báo cáo kết quả  
public static void results (int totalGame, int totalGuess, int best, double avg) {  
  
 System.*out*.println("");  
 System.*out*.println("Overall results:");  
 System.*out*.println(" total games = " + totalGame);  
 System.*out*.println(" total guesses = " + totalGuess);  
 System.*out*.println(" guesses/game = " + avg);  
 System.*out*.println(" best game = " + best);  
  
}

|  |
| --- |
| import java.util.Scanner; |
|  | import java.math.\*; |
|  | // Ari Madian |
|  | // CSC 142 |
|  | // Ravi Gandham |
|  | // 10/20/18 |
|  |  |
|  | public class Gradanator { |
|  |  |
|  | private static final Scanner scanner = new Scanner(System.in); |
|  |  |
|  | public static void main(String[] args) { |
|  | System.out.println("This program read exam/homework scores\nand reports your overall course grade.\n"); |
|  | double weightedMidtermScore = midtermOrFinal("Midterm:"); |
|  | double weightedFinalScore = midtermOrFinal("Final:"); |
|  | double weightedHomeWorkScore = homework(); |
|  | double totalGrade = weightedMidtermScore + weightedFinalScore + weightedHomeWorkScore; |
|  |  |
|  | printFinalGrades(totalGrade); |
|  | } |
|  |  |
|  | /\* |
|  | <midtermOrFinal> - Gets all relevant values, calculates weighted midterm or final score. |
|  | <param> String mof - Either "Midterm:" or "Final:" tells the method what to print for the section header. |
|  | <returns> double - The rounded, weighted, score for the given section. |
|  | \*/ |
|  | private static double midtermOrFinal(String mof){ |
|  | System.out.println(mof); |
|  |  |
|  | // Get Starting Values |
|  | System.out.print("Weight (0-100)? "); double Weight = Double.parseDouble(scanner.nextLine()); |
|  | System.out.print("Score Earned? "); double ScoreEarned = Double.parseDouble(scanner.nextLine()); |
|  | System.out.print("Were Scores Shifted (1=yes, 2=no)? "); int ScoresShifted = Integer.parseInt(scanner.nextLine()); |
|  |  |
|  | double weightScalar = Weight \* .01; |
|  |  |
|  | // If Scores Are Shifted, Get Shift Amount |
|  | if (ScoresShifted == 1){ |
|  | System.out.print("Shift Amount? "); int shiftAmount = Integer.parseInt(scanner.nextLine()); |
|  | if ((shiftAmount + ScoreEarned) >= 100) { // Check to see if shifted amount is > 100, if yes, = 100. |
|  | ScoreEarned = 100; |
|  | } else { |
|  | ScoreEarned += ScoresShifted; |
|  | } |
|  | } |
|  |  |
|  | // Print totals |
|  | printSectionFinal(ScoreEarned, 100.0, weightScalar); |
|  |  |
|  | System.out.println(); |
|  | return round(ScoreEarned \* weightScalar); |
|  | } |
|  |  |
|  | /\* |
|  | <homework> - Gets all relevant values, calculates weighted homework score. |
|  | <return> double - The rounded, weighted, homework section score. |
|  | \*/ |
|  | private static double homework(){ |
|  | double earnedPoints = 0.0; |
|  | double totalPoints = 0.0; |
|  |  |
|  | // Get Starting Values |
|  | System.out.println("Homework:"); |
|  | System.out.print("Weight (0-100)? "); double Weight = Double.parseDouble(scanner.nextLine()); |
|  | System.out.print("Number Of Assignments? "); int numAssignments = Integer.parseInt(scanner.nextLine()); |
|  | double weightScalar = Weight \* .01; |
|  |  |
|  | // For number of assignments given, get assignment score info. |
|  | for (int assignmentNum = 1; assignmentNum <= numAssignments; assignmentNum++){ |
|  | // Get Input |
|  | System.out.print("Assignment " + assignmentNum + " score and max? "); String scoreAndMax= scanner.nextLine();//điểm đạt và điểm max |
|  |  |
|  | // Extract Values From String |
|  | double earned = Double.parseDouble(scoreAndMax.substring(0, 2)); |
|  | double total = Double.parseDouble(scoreAndMax.substring(scoreAndMax.length() - 2, scoreAndMax.length())); |
|  |  |
|  | // Add Points Up |
|  | earnedPoints += earned; |
|  | totalPoints += total; |
|  | } |
|  |  |
|  | // Section Points |
|  | System.out.print("How many sections did you attend? "); int sectionsAttended = Integer.parseInt(scanner.nextLine()); |
|  | if ((sectionsAttended \* 3) >= 20){ // Cap points if greater than max allowed |
|  | earnedPoints += 20; |
|  | System.out.println("Section Points = 20 / 20"); |
|  | } else { |
|  | earnedPoints += sectionsAttended \* 3; |
|  | System.out.println("Section Points = " + sectionsAttended \* 3 + " / 20"); |
|  | } |
|  | totalPoints += 20; |
|  |  |
|  | printSectionFinal(earnedPoints, totalPoints, weightScalar); |
|  |  |
|  | System.out.println(); |
|  | return round((earnedPoints / totalPoints) \* Weight); |
|  | } |
|  |  |
|  | /\* |
|  | <round> - Rounds a double to the first decimal point. |
|  | <param> double value - The value to round |
|  | <returns> double - The rounded value |
|  | \*/ |
|  | private static double round(double value) { |
|  | BigDecimal bd = new BigDecimal(value); |
|  | bd = bd.setScale(1, RoundingMode.HALF\_UP); |
|  | return bd.doubleValue(); |
|  | } |
|  |  |
|  | /\* |
|  | <printSectionFinal> - Prints the total points earned out of points possible, and the weighted score for the section. |
|  | <param> double earnedPoints - The number of points earned for the given section. |
|  | <param> double totalPoints - The total number of points possible for the given section. |
|  | <param> double scalarValue - The section's weighing scalar value. |
|  | \*/ |
|  | private static void printSectionFinal(double earnedPoints, double totalPoints, double scalarValue){ |
|  | double oneHunTScalar = 100 \* scalarValue; |
|  | System.out.println("Total Points = " + round(earnedPoints) + " / " + (int)totalPoints); |
|  | System.out.println("Weighted Score = " + round((earnedPoints \* oneHunTScalar) / totalPoints) + " / " + (int)oneHunTScalar); |
|  |  |
|  | } |
|  |  |
|  | /\* |
|  | <printFinalGrades> - Prints the overall grade, rounded to the first decimal point, minimum GPA, and a custom message. |
|  | <param> double grade - The student's total, rounded, grade. |
|  | \*/ |
|  | private static void printFinalGrades(double grade){ |
|  | System.out.println("Overall Percentage = " + grade); |
|  | if (grade >= 85.0){ System.out.println("Your grade will be at least: 3.0\nGood Job!"); |
|  | } else if (grade >= 75.0){ System.out.println("Your grade will be at least: 2.0\nNot too bad..."); |
|  | } else if (grade >= 60.0){ System.out.println("Your grade will be at least: 0.7\nYou should probably study more next time."); |
|  | } else { System.out.println("Your grade will be at least: 0.0\nReally?"); } |
|  | } |
|  | } |

public static void main(String[] args) { giveIntro(); Scanner console = new Scanner(System.in); int Grade ; double weightedScore = weightedScore(weight, score, curveNumber); double weightedScore = weightedScore2(weight2, sections, sumScore); } // a welcome message to start the program public static void giveIntro() { System.out.println("This program accepts your homework and exam"); System.out.println("scores as input and,computes your grade in"); System.out.println("the course or indicates what grade you need"); System.out.println("to earn on the final exam."); System.out.println(); } // asks about student's homeworks and returns weighted HW score public static void getHomeworkScores() { System.out.print("What is its weight (0-100)?"); double weight = console.nextInt(); } public static double homework(int weight) { return 0.0; } // asks about student's exam and returns weighted exam score public static double exam(Scanner console, int number, int weight) { return 0.0; } // helper method to compute and print a weighted category score public static double weightedScore(int weight, int earned, int possible) { return 0.0; } // returns the given double value rounded to the nearest hundredth public static double round2(double number) { return Math.round(number \* 100.0) / 100.0; } }  
  
Source https://www.physicsforums.com/threads/grades-java-stuck.681321/