|  |  |
| --- | --- |
|  | ***Patuakhali Science and Technology University*** |

Assignment on

***“***Linag book chapter 5 solve***”***

Course Code: CCE-122

Course Title: Object Oriented Programming

Level - I; Semester - II

|  |
| --- |
| **Submitted By**  **Name: M.D. Sakibul Islam Shovon**  **ID:** 2302056  **REG:** 11834  **Session:** 2023-2024  Faculty of Computer Science and Engineering |

|  |
| --- |
| **Submitted To**  **Prof. Dr. Md. Samsuzzaman**  Professor of Computer and Communication Engineering Department  Faculty of Computer Science and Engineering |

**Linag book chapter 5 solve**

**5.1 (Pass or fail) Write a program that prompts a student to enter a Java score. If the score is greater or equal to 60, display “you pass the exam”; otherwise, display “you don’t pass the exam”. Your program ends with input -1. Here is a sample run:**

Solution:

import java.util.Scanner;

public class PassOrFail {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int score;

        while (true) {

            System.out.print("Enter Java score (-1 to exit): ");

            score = input.nextInt();

            if (score == -1) {

                System.out.println("Program ended.");

                break;

            }

            if (score >= 60) {

                System.out.println("You pass the exam");

            } else {

                System.out.println("You don’t pass the exam");

            }

        }

    }

}

**5.2 (Repeat multiplications) Listing 5.4, SubtractionQuizLoop.java, generates five random subtraction questions. Revise the program to generate ten random multiplication ques tions for two integers between 1 and 12. Display the correct count and test time.**

Solution:

import java.util.Scanner;

public class RepeatMultiplications {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

int correctCount = 0;

long startTime = System.currentTimeMillis();

for (int i = 0; i < 10; i++) {

int num1 = 1 + (int)(Math.random() \* 12);

int num2 = 1 + (int)(Math.random() \* 12);

System.out.print("What is " + num1 + " \* " + num2 + "? ");

int answer = input.nextInt();

if (answer == num1 \* num2) {

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

correctCount++;

} else {

System.out.println("Wrong. The answer is " + (num1 \* num2));

}

}

long endTime = System.currentTimeMillis();

System.out.println("Correct count: " + correctCount);

System.out.println("Test time: " + ((endTime - startTime) / 1000) + " seconds");

}

}

5.3 (Conversion from Cº to Fº) Write a program that displays the following table (note that farenheit = celsius \* 9/5 + 32): Celsius Fahrenheit 0 32.0 2 35.6 ... 98 208.4 100 212.0

Solution:

public class CelsiusToFahrenheit {

public static void main(String[] args) {

System.out.println("Celsius\tFahrenheit");

for (int celsius = 0; celsius <= 100; celsius += 2) {

double fahrenheit = celsius \* 9.0 / 5 + 32;

System.out.printf("%d\t\t%.1f%n", celsius, fahrenheit);

}

}

}

5.4 (Conversion from inch to centimeter) Write a program that displays the following table (note that 1 inch is 2.54 centimeters): Inches Centimetres 1 2.54 2 5.08 ... 9 22.86 10 25.4

Solution:

public class InchesToCentimeters {

public static void main(String[] args) {

System.out.println("Inches\tCentimeters");

for (int inches = 1; inches <= 10; inches++) {

double cm = inches \* 2.54;

System.out.printf("%d\t\t%.2f%n", inches, cm);

}

}

}

5.5 (Conversion from Cº to Fº and Fº to Cº) Write a program that displays the following two tables side by side: Celsius Fahrenheit | Fahrenheit Celsius 0 32.000 | 20 −6.667 2 35.600 | 25 −3.889 ... 98 208.400 | 265 129.444 100 212.000 | 270 132.222

Solution:

public class TempConversion {

public static void main(String[] args) {

System.out.println("Celsius Fahrenheit | Fahrenheit Celsius");

int celsius = 0;

int fahrenheit = 20;

while (celsius <= 100 && fahrenheit <= 270) {

double f = celsius \* 9.0 / 5 + 32;

double c = (fahrenheit - 32) \* 5.0 / 9;

System.out.printf("%-8d%-11.3f| %-11d%.3f%n", celsius, f, fahrenheit, c);

celsius += 2;

fahrenheit += 5;

}

}

}

5.6 (Conversion from square meter to ping) Write a program that displays the following two tables side by side (note that 1 ping = 3.305 square meters): Ping Square meter | Square meter Ping 10 33.050 | 30 9.077 15 49.575 | 35 10.590 ... 75 247.875 | 95 28.744 80 264.400 | 100 30.257

Solution:

public class SquareMeterToPing {

public static void main(String[] args) {

System.out.println("Ping Square meter | Square meter Ping");

int ping = 10;

int sqm = 30;

while (ping <= 80 && sqm <= 100) {

double sqmFromPing = ping \* 3.305;

double pingFromSqm = sqm / 3.305;

System.out.printf("%-6d%-14.3f| %-14d%.3f%n", ping, sqmFromPing, sqm, pingFromSqm);

ping += 5;

sqm += 5;

}

}

}

5.7 (Financial application: compute future tuition) Suppose that the tuition for a uni versity is $10,000 this year and increases 6% every year. In one year, the tuition will be $10,600. Write a program that computes the tuition in ten years and the total cost of four years’ worth of tuition after the tenth year.

Soution:

public class FutureTuition {

public static void main(String[] args) {

double tuition = 10000;

for (int year = 1; year <= 10; year++) {

tuition \*= 1.06;

}

System.out.printf("Tuition in 10 years: $%.2f%n", tuition);

double totalCost = 0;

for (int i = 0; i < 4; i++) {

totalCost += tuition;

tuition \*= 1.06;

}

System.out.printf("Total cost for 4 years after 10 years: $%.2f%n", totalCost);

}

}

5.8 (Find the highest score) Write a program that prompts the user to enter the number of students and each student’s name and score, and finally displays the name of the student with the highest score. Use the next() method in the Scanner class to read a name, rather than using the nextLine() method.

Solution:

import java.util.Scanner;

public class HighestScore {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter number of students: ");

int num = input.nextInt();

String topStudent = "";

int highestScore = -1;

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

System.out.print("Enter name: ");

String name = input.next();

System.out.print("Enter score: ");

int score = input.nextInt();

if (score > highestScore) {

highestScore = score;

topStudent = name;

}

}

System.out.println("Highest score: " + topStudent);

}

}

5.9 (Find the two lowest scores) Write a program that prompts the user to enter the number of students and each student’s name and score, and finally displays the names of the students with the lowest and second-lowest scores.

Solution:

import java.util.Scanner;

public class TwoLowestScores {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter number of students: ");

int num = input.nextInt();

String low1Name = "", low2Name = "";

int low1 = Integer.MAX\_VALUE, low2 = Integer.MAX\_VALUE;

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

System.out.print("Enter name: ");

String name = input.next();

System.out.print("Enter score: ");

int score = input.nextInt();

if (score < low1) {

low2 = low1;

low2Name = low1Name;

low1 = score;

low1Name = name;

} else if (score < low2) {

low2 = score;

low2Name = name;

}

}

System.out.println("Lowest score: " + low1Name);

System.out.println("Second lowest score: " + low2Name);

}

}

5.10 (Find numbers divisible by 3 and 4) Write a program that displays all the numbers from 100 to 1,000, ten per line, that are divisible by 3 and 4. Numbers are separated by exactly one space.

Solution:

public class DivisibleBy3And4 {

public static void main(String[] args) {

int count = 0;

for (int i = 100; i <= 1000; i++) {

if (i % 3 == 0 && i % 4 == 0) {

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

count++;

if (count % 10 == 0) System.out.println();

}

}

}

}

5.11 (Find numbers divisible by 3 or 4, but not both) Write a program that displays all the numbers from 100 to 200, ten per line, that are divisible by 3 or 4, but not both. Numbers are separated by exactly one space.

Solution:

public class DivisibleBy3Or4NotBoth {

public static void main(String[] args) {

int count = 0;

for (int i = 100; i <= 200; i++) {

if ((i % 3 == 0) ^ (i % 4 == 0)) {

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

count++;

if (count % 10 == 0) System.out.println();

}

}

}

}

5.12 (Find the smallest n such that n3 7 12,000) Use a while loop to find the smallest integer n such that n3 is greater than 12,000.

Solution:

public class SmallestNCube {

public static void main(String[] args) {

int n = 1;

while (n \* n \* n <= 12000) {

n++;

}

System.out.println("Smallest n: " + n);

}

}

5.13 (Find the largest n such that n2 6 12,000) Use a while loop to find the largest integer n such that n2 is less than 12,000.

Solution:

public class LargestNSquare {

public static void main(String[] args) {

int n = 1;

while (n \* n < 12000) {

n++;

}

System.out.println("Largest n: " + (n - 1));

}

}

5.14 (Compute the greatest common divisor) Another solution for Listing 5.9 to find the greatest common divisor of two integers n1 and n2 is as follows: First find d to be the minimum of n1 and n2, then check whether d, d–1, d–2, …, 2, or 1 is a divi sor for both n1 and n2 in this order. The first such common divisor is the greatest common divisor for n1 and n2. Write a program that prompts the user to enter two positive integers and displays the gcd.

Solution:

import java.util.Scanner;

public class GCD {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter first number: ");

int n1 = input.nextInt();

System.out.print("Enter second number: ");

int n2 = input.nextInt();

int d = Math.min(n1, n2);

while (d > 0) {

if (n1 % d == 0 && n2 % d == 0) {

break;

}

d--;

}

System.out.println("GCD is " + d);

}

}

5.15 (Display the ASCII character table) Write a program that prints the characters in the ASCII character table from ! to ~. Display 10 characters per line. The ASCII table is given in Appendix B. Characters are separated by exactly one space.

Solution:

public class AsciiTable {

public static void main(String[] args) {

int count = 0;

for (char ch = '!'; ch <= '~'; ch++) {

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

count++;

if (count % 10 == 0) System.out.println();

}

}

}