Chapter 5

**5.7 Values of x**

a) 7.5  
b) 7.0  
c) 0.0  
d) 0.0  
e) 6.4  
f) -6.0  
g) -14.0

**5.8 Parking Charges**

java

Copy

import java.util.Scanner;

public class ParkingGarage {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

double total = 0.0;

System.out.println("Enter hours parked for each customer (negative to quit):");

while (true) {

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

double hours = input.nextDouble();

if (hours < 0) break;

double charge = calculateCharges(hours);

total += charge;

System.out.printf("Charge: $%.2f, Total: $%.2f%n", charge, total);

}

input.close();

}

public static double calculateCharges(double hours) {

double charge = 2.0;

if (hours > 3) {

double excess = Math.ceil(hours - 3);

charge += excess \* 0.5;

}

return Math.min(charge, 10.0);

}

}

**5.9 Rounding Numbers**

java

Copy

import java.util.Scanner;

public class Rounding {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.println("Enter numbers:");

while (input.hasNextDouble()) {

double x = input.nextDouble();

int y = (int) Math.floor(x + 0.5);

System.out.printf("Original: %.2f, Rounded: %d%n", x, y);

}

input.close();

}

}

**5.10 Rounding to Specific Decimals**

java

Copy

import java.util.Scanner;

public class DecimalRounding {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

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

double x = input.nextDouble();

System.out.printf("Original: %f%n", x);

System.out.printf("Rounded to integer: %d%n", roundToInteger(x));

System.out.printf("Rounded to tenths: %.1f%n", roundToTenths(x));

System.out.printf("Rounded to hundredths: %.2f%n", roundToHundredths(x));

System.out.printf("Rounded to thousandths: %.3f%n", roundToThousandths(x));

input.close();

}

public static int roundToInteger(double number) {

return (int) Math.floor(number + 0.5);

}

public static double roundToTenths(double number) {

return Math.floor(number \* 10 + 0.5) / 10;

}

public static double roundToHundredths(double number) {

return Math.floor(number \* 100 + 0.5) / 100;

}

public static double roundToThousandths(double number) {

return Math.floor(number \* 1000 + 0.5) / 1000;

}

}

// 5.12

public static double triangleArea(double base, double height) {

return 0.5 \* base \* height;

}

// 5.13

public static boolean isPerfectSquare(int num) {

int sqrt = (int) Math.sqrt(num);

return sqrt \* sqrt == num;

}

// 5.14

public static double celsiusToFahrenheit(double celsius) {

return (celsius \* 9/5) + 32;

}

// 5.15

public static void printSquareAndCube(int number) {

System.out.println("Square: " + (number \* number));

System.out.println("Cube: " + (number \* number \* number));

}

// 5.16

public static boolean isDivisibleBy3And5(int number) {

return number % 3 == 0 && number % 5 == 0;

}

// 5.17

public static boolean isPrime(int num) {

if (num <= 1) return false;

for (int i = 2; i <= Math.sqrt(num); i++) {

if (num % i == 0) return false;

}

return true;

}

// 5.18

public static long roundNumber(double num) {

return Math.round(num);

}

// 5.19

public static void printEvenNumbers1To100() {

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

if (i % 2 == 0) {

System.out.println(i);

}

}

}

// 5.20

public static double simpleInterest(double principal, double rate, double time) {

return (principal \* rate \* time) / 100;

}

// 5.21

public static int maxOfThree(int a, int b, int c) {

return Math.max(a, Math.max(b, c));

}

// 5.22

public static int absoluteValue(int num) {

return Math.abs(num);

}

// 5.23

public static int generateRandomNumber() {

return (int)(Math.random() \* 100) + 1;

}

// 5.24

public static double rectangleArea(double length, double width) {

return length \* width;

}

// 5.25

public static long factorial(int num) {

long result = 1;

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

result \*= i;

}

return result;

}

// 5.26

public static boolean isLeapYear(int year) {

return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);

}

// 5.27

public static int sumOfDigits(int num) {

int sum = 0;

while (num > 0) {

sum += num % 10;

num /= 10;

}

return sum;

}

// 5.28

public static int reverseNumber(int num) {

int reversed = 0;

while (num != 0) {

int digit = num % 10;

reversed = reversed \* 10 + digit;

num /= 10;

}

return reversed;

}

// 5.29

public static void multiplicationTable(int number) {

for (int i = 1; i <= 12; i++) {

System.out.println(number + " x " + i + " = " + (number \* i));

}

}