\* Hello world program

public class HelloWorld {

public static void main(String[] args) {

System.out.println("Hello, World!");

}

}

\* Print numbers from 1-10

public class PrintNumbers {

public static void main(String[] args) {

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

System.out.println(i);

}

}

}

\* Print array elements

public class PrintArray {

public static void main(String[] args) {

int[] numbers = {10, 20, 30, 40, 50};

for (int number : numbers) {

System.out.println(number);

}

}

}

\* Input array elements

import java.util.Scanner;

public class InputArray {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the size of the array: ");

int size = scanner.nextInt();

int[] numbers = new int[size];

System.out.println("Enter " + size + " elements:");

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

numbers[i] = scanner.nextInt();

}

System.out.println("You entered:");

for (int number : numbers) {

System.out.println(number);

}

}

}

\* Define method to print array elements

public class MethodPrintArray {

public static void printArray(int[] array) {

for (int number : array) {

System.out.println(number);

}

}

public static void main(String[] args) {

int[] numbers = {10, 20, 30, 40, 50};

printArray(numbers);

}

}

\* Define method to Input array elements

import java.util.Scanner;

public class MethodInputArray {

public static int[] inputArray(int size) {

Scanner scanner = new Scanner(System.in);

int[] array = new int[size];

System.out.println("Enter " + size + " elements:");

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

array[i] = scanner.nextInt();

}

return array;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the size of the array: ");

int size = scanner.nextInt();

int[] numbers = inputArray(size);

System.out.println("You entered:");

for (int number : numbers) {

System.out.println(number);

}

}

}

\* Array of objects (Students)

class Student {

String name;

int id;

Student(String name, int id) {

this.name = name;

this.id = id;

}

void display() {

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

}

}

public class ArrayOfObjects {

public static void main(String[] args) {

Student[] students = new Student[3];

students[0] = new Student("Ahmed", 1);

students[1] = new Student("Fatima", 2);

students[2] = new Student("Ali", 3);

for (Student student : students) {

student.display();

}

}

}

Homework

Create a github account and upload lecture programs to your account

1. R-1.1
2. Write a short Java method, inputAllBaseTypes, that inputs a different value of each base type from the standard input device and prints it back to the standard output device.import java.util.Scanner;

public class InputAllBaseTypes {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter an integer: ");

int intValue = scanner.nextInt();

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

double doubleValue = scanner.nextDouble();

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

float floatValue = scanner.nextFloat();

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

char charValue = scanner.next().charAt(0);

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

scanner.nextLine(); // Clear buffer

String stringValue = scanner.nextLine();

System.out.println("You entered:");

System.out.println("Integer: " + intValue);

System.out.println("Double: " + doubleValue);

System.out.println("Float: " + floatValue);

System.out.println("Character: " + charValue);

System.out.println("String: " + stringValue);

}

}

2 R-1.2

Suppose that we create an array A of GameEntry objects, which has an integer scores field, and we clone A and store the result in an array B. If we then immediately set A[4].score equal to 550, what is the score value of the GameEntry object referenced by B[4]?class GameEntry {

int score;

GameEntry(int score) {

this.score = score;

}

}

public class GameEntryArray {

public static void main(String[] args) {

GameEntry[] A = { new GameEntry(10), new GameEntry(20), new GameEntry(30), new GameEntry(40), new GameEntry(50) };

GameEntry[] B = A.clone();

B[4].score = 250;

System.out.println("B[4] score: " + B[4].score);

}

}

3 R-1.3

Write a short Java method, isMultiple, that takes two long values, n and m, and returns true if and only if n is a multiple of m, that is, n = mi for some integer i.public class IsMultiple {

public static boolean isMultiple(long n, long m) {

return n % m == 0;

}

public static void main(String[] args) {

System.out.println(isMultiple(10, 2)); // true

System.out.println(isMultiple(10, 3)); // false

}

}

4 R-1.4 Write a short Java method, isEven, that takes an int i and returns true if and only if i is even. Your method cannot use the multiplication, modulus, or division operators, however.public class IsEven {

public static boolean isEven(int i) {

return (i & 1) == 0; // Check the least significant bit

}

public static void main(String[] args) {

System.out.println(isEven(10)); // true

System.out.println(isEven(7)); // false

}

}

5 R-1.5 Write a short Java method that takes an integer n and returns the sum of all positive integers less than or equal to n.public class SumToN {

public static int sum(int n) {

return n \* (n + 1) / 2; // Formula for sum of first n numbers

}

public static void main(String[] args) {

System.out.println(sum(5)); // 15

}

}

6 R-1.6

Write a short Java method that takes an integer n and returns the sum of all the odd positive integers less than or equal to n.public class SumOddToN {

public static int sumOdd(int n) {

int sum = 0;

for (int i = 1; i <= n; i += 2) {

sum += i;

}

return sum;

}

public static void main(String[] args) {

System.out.println(sumOdd(10)); // 25

}

}

7 R-1.7

Write a short Java method that takes an integer n and returns the sum of the squares of all positive integers less than or equal to n.public class SumSquaresToN {

public static int sumSquares(int n) {

int sum = 0;

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

sum += i \* i;

}

return sum;

}

public static void main(String[] args) {

System.out.println(sumSquares(3)); // 14

}

}

8 R-1.8

Write a short Java method that counts the number of vowels in a given character string.public class CountVowels {

public static int countVowels(String s) {

int count = 0;

for (char c : s.toLowerCase().toCharArray()) {

if ("aeiou".indexOf(c) != -1) {

count++;

}

}

return count;

}

public static void main(String[] args) {

System.out.println(countVowels("Hello World")); // 3

}

}

9 R-1.9

Write a short Java method that uses a StringBuilder instance to remove all the punctuation from a string s storing a sentence, for example, transforming the string "Let’s try, Mike!" to "Lets try Mike".public class RemovePunctuation {

public static String removePunctuation(String s) {

return s.replaceAll("\\p{Punct}", "");

}

public static void main(String[] args) {

System.out.println(removePunctuation("Let's try, Mike!")); // "Lets try Mike"

}

}

10 R-1.10

Write a Java class, Flower, that has three instance variables of type String, int, and float, which respectively represent the name of the flower, its number of petals, and price. Your class must include a constructor method that initializes each variable to an appropriate value, and your class should include methods for setting the value of each type, and getting the value of each type.class Flower {

private String name;

private int numberOfPetals;

private float price;

// Constructor to initialize variables

public Flower(String name, int numberOfPetals, float price) {

this.name = name;

this.numberOfPetals = numberOfPetals;

this.price = price;

}

// Setter methods

public void setName(String name) {

this.name = name;

}

public void setNumberOfPetals(int numberOfPetals) {

this.numberOfPetals = numberOfPetals;

}

public void setPrice(float price) {

this.price = price;

}

// Getter methods

public String getName() {

return name;

}

public int getNumberOfPetals() {

return numberOfPetals;

}

public float getPrice() {

return price;

}

// Main method to test the class

public static void main(String[] args) {

Flower flower = new Flower("Rose", 7, 2.5f);

System.out.println("Name: " + flower.getName());

System.out.println("Number of Petals: " + flower.getNumberOfPetals());

System.out.println("Price: " + flower.getPrice());

// Update values

flower.setName("Lily");

flower.setNumberOfPetals(5);

flower.setPrice(3.0f);

System.out.println("Updated Name: " + flower.getName());

System.out.println("Updated Number of Petals: " + flower.getNumberOfPetals());

System.out.println("Updated Price: " + flower.getPrice());

}

}

11 R-1.11

Modify the CreditCard class from Code Fragment 1.5 to include a method that updates the credit limit.class CreditCard {

private double creditLimit;

public CreditCard(double creditLimit) {

this.creditLimit = creditLimit;

}

public void updateCreditLimit(double newLimit) {

this.creditLimit = newLimit;

}

public double getCreditLimit() {

return creditLimit;

}

public static void main(String[] args) {

CreditCard card = new CreditCard(5000.0);

System.out.println("Old Credit Limit: " + card.getCreditLimit());

card.updateCreditLimit(7500.0);

System.out.println("Updated Credit Limit: " + card.getCreditLimit());

}

}

12 R-1.12

Modify the CreditCard class from Code Fragment 1.5 so that it ignores any request to process a negative payment amount.class CreditCard {

private double balance;

public CreditCard() {

this.balance = 0.0;

}

public void processPayment(double amount) {

if (amount < 0) {

System.out.println("Negative payment ignored.");

} else {

balance -= amount;

System.out.println("Payment processed. New balance: " + balance);

}

}

public static void main(String[] args) {

CreditCard card = new CreditCard();

card.processPayment(100); // Valid payment

card.processPayment(-50); // Invalid payment

}

}

13 R-1.13

Modify the declaration of the first for loop in the main method in Code Fragment 1.6 so that its charges will cause exactly one of the three credit cards to attempt to go over its credit limit. Which credit card is it?class CreditCard {

private double balance;

private double creditLimit;

public CreditCard(double creditLimit) {

this.balance = 0.0;

this.creditLimit = creditLimit;

}

public boolean charge(double amount) {

if (balance + amount > creditLimit) {

System.out.println("Charge denied. Exceeds credit limit.");

return false;

} else {

balance += amount;

System.out.println("Charge approved. New balance: " + balance);

return true;

}

}

public static void main(String[] args) {

CreditCard[] cards = {

new CreditCard(1000), // Card 1

new CreditCard(2000), // Card 2

new CreditCard(3000) // Card 3

};

// Modify the loop to exceed the limit for the second card

double[] charges = {500, 2500, 1000}; // Charges per card

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

System.out.println("Processing charges for Card " + (i + 1));

cards[i].charge(charges[i]);

}

}

}