**Java programming**

**Exercise 1 of 16**

**KARAGO ANNETTE**

**23/06380**

**Instructions:**

All programs should be written, and linked to an online repository like GitHub.

A video to get you started with GitHub has been posted on Moodle.

After completing your assignment, post the link on the link on Moodle. An instructor will follow the posted link to access and grade your work.

Note that: Your program should always be well-commented. At the top of your source code file, you should write a short description of what your program does and add other comments to help in explaining your code.

All of your variables should be given a deceptive name. Avoid giving your variables names like a, b, I, x, y etc.

In case you copy your friend's work, you both get a Zero (0).

**Section 1:**

1. Explain the differences between primitive and reference data types.

**Primitive types**: These are predefined by the language and named by a keyword. Examples include int, char, Boolean, etc. They store actual values.

**Reference types**: These refer to objects or arrays and store the memory address of the object they refer to. Examples include String, Arrays, Classes, etc.

1. Define the scope of a variable (hint: local and global variable)

**Local variable**: Declared inside a method or a block and can only be accessed within that method/block.

**Global variable (or Instance variable)**: Declared inside a class but outside any method and can be accessed by all methods in that class.

1. Why is initialization of variables required?

Initialization ensures that variables hold a valid value before being used in computations. Using uninitialized variables can cause runtime errors or unpredictable behaviour.

1. Differentiate between static, instance and local variables.

**Static variable**: Belongs to the class and shared among all instances of that class.

**Instance variable**: Unique to each instance of a class.

**Local variable**: Declared inside a method and exists only within the scope of that method.

1. Differentiate between widening and narrowing casting in java.

**Widening casting**: Converts a smaller type to a larger type (e.g., int to long). It’s automatic and does not cause data loss.

**Narrowing casting**: Converts a larger type to a smaller type (e.g., double to int). It requires explicit casting and may cause data loss.

1. the following table shows data type, its size, default value and the range. Filling in the missing values.

|  |  |  |  |
| --- | --- | --- | --- |
| **TYPE** | **SIZE (IN BYTES)** | **DEFAULT** | **RANGE** |
| boolean | 1 bit | false | true, false |
| Char | 2 | \u0000 | ‘\0000’ to ‘\ffff’ |
| Byte | 1 | 0 | -128 to 127 |
| Short | 2 | 0 | -215 to +215-1 |
| Int | 4 | 0 | -2^31 to 2^31 - 1 |
| Long | 8 | 0L | - -2^63 to 2^63 - 1 |
| Float | 4 | 00.0f | 1.4E-45 to 3.4028235E+38 |
| Double | 8 | 0.0d | -1.8E+308 to +1.8E+308 |

1. Define class as used in OOP

A **class** in Object-Oriented Programming is a blueprint for creating objects. It encapsulates data for the object and methods to manipulate that data.

1. Explain the importance of classes in Java programming.

Classes enable modularity and reusability. They allow data and methods to be encapsulated and used as units, reducing complexity and improving maintainability.

Section 2:

1. Write a Java program that asks the user to enter their sur name and current age then print the number of characters of their sir name and even or odd depending on their age number.

Example of Expected result:

If sir name is Saruni and age is 29, output will be;

then the number of characters is 6.

Your current age is an odd number

import java.util.Scanner;

public class SurnameAgeChecker {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

// Get user's surname and age

System.out.print("Enter your surname: ");

String surname = input.nextLine();

System.out.print("Enter your age: ");

int age = input.nextInt();

// Print number of characters in surname

System.out.println("The number of characters in your surname is " + surname.length());

// Check if age is odd or even

if (age % 2 == 0) {

System.out.println("Your current age is an even number.");

} else {

System.out.println("Your current age is an odd number.");

}

}

}

1. Write Java program to ask student to enter the marks of the five units they did last semester, compute the average and display it on the screen. (Average should be given in two decimal places).

import java.util.Scanner;

public class StudentAverage {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

double totalMarks = 0;

// Input marks for five units

for (int i = 1; i <= 5; i++) {

System.out.print("Enter marks for unit " + i + ": ");

double marks = input.nextDouble();

totalMarks += marks;

}

// Calculate and display average

double average = totalMarks / 5;

System.out.printf("The average marks are: %.2f", average);

}

}

1. Write a program that will help kids learn divisibly test of numbers of integers. The program should check whether the given integer is divisible by integers in the range of 0-9. For example, if a number (955) is divisible by five, the program should print, the number is divisible by 5 because it ends with a 5, and 900 is divisible by 5 because it ends with a 0(zero).

import java.util.Scanner;

public class DivisibilityTest {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

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

int number = input.nextInt();

for (int i = 1; i <= 9; i++) {

if (number % i == 0) {

System.out.println(number + " is divisible by " + i);

}

}

}

}

1. Write a Java program to display all the multiples of 2, 3 and 7 within the range 71 to 150.

public class Multiples {

public static void main(String[] args) {

System.out.println("Multiples of 2, 3, and 7 between 71 and 150:");

for (int i = 71; i <= 150; i++) {

if (i % 2 == 0 || i % 3 == 0 || i % 7 == 0) {

System.out.println(i);

}

}

}

}

1. Create a calculator using java to help user perform the basic operations (+, -, \* and /).
   1. User should be asked to enter a number, then an operation, the program computes the operation and display the output to the computer screen.

import java.util.Scanner;

public class Calculator {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

// Input numbers and operation

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

double num1 = input.nextDouble();

System.out.print("Enter an operation (+, -, \*, /): ");

char operation = input.next().charAt(0);

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

double num2 = input.nextDouble();

// Perform calculation and display result

switch (operation) {

case '+':

System.out.println("Result: " + (num1 + num2));

break;

case '-':

System.out.println("Result: " + (num1 - num2));

break;

case '\*':

System.out.println("Result: " + (num1 \* num2));

break;

case '/':

if (num2 != 0) {

System.out.println("Result: " + (num1 / num2));

} else {

System.out.println("Error: Division by zero.");

}

break;

default:

System.out.println("Invalid operation.");

}

}

}