Java Assignment

22/02935- Mungai John

***Section 1***

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

In programming, the terms "primitive" and "reference" data types are used to distinguish different categories of data types based on their behavior and memory management. Here are the main differences between these two types:

* + Primitive Data Types: Primitive data types are basic data types that are built into the programming language. They represent simple values and are not composed of other data types. Examples of primitive data types include integers, floating-point numbers, characters, booleans, and so on.
  + Reference Data Types: Reference data types, also known as objects or reference types, are more complex and can be composed of multiple values and methods. They are typically created using classes or structures defined in the programming language or custom-defined by the programmer.

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

The scope of a variable refers to the region of a program where the variable is accessible or visible.

Local Variables:

* Local variables are declared within a specific block of code, such as a function or a loop.
* They are only accessible within the block in which they are defined, and their lifetime is limited to that block.

Global Variables:

* Global variables are declared outside of any specific block or function, typically at the top of the program or in a global scope.
* They are accessible and can be used throughout the entire program, from any part of the code, including different functions, classes, or files.
* Global variables have a longer lifetime compared to local variables and persist throughout the execution of the program.

1. Why is initialization of variables required

Preventing Undefined Behavior: When a variable is declared but not initialized, its value is undefined, meaning it can contain any arbitrary value that happened to be in memory at that location.

Preventing Undefined Behavior: When a variable is declared but not initialized, its value is undefined, meaning it can contain any arbitrary value that happened to be in memory at that location.

Avoiding Garbage/Residual Values: Sometimes, variables may hold garbage or residual values from previous operations or memory allocations

1. Differentiate between static, instance and local variables.

Static Variables:

* Static variables are declared using the **static** keyword in most programming languages.
* They are associated with the class or the file in which they are defined, rather than with instances (objects) of the class.

Instance Variables:

* Instance variables, also known as non-static variables or member variables, are declared within a class but outside any method or constructor.
* Each instance of a class has its own set of instance variables, which are independent of other instances.
* Instance variables are created and initialized when an object (instance) of the class is created.

Local Variables:

* Local variables are declared within a specific block of code, such as a method, function, or loop.
* They are only accessible within the block in which they are defined.
* Local variables have a limited lifetime and are created when the block is entered and destroyed when the block is exited.

1. Differentiate between widening and narrowing casting in java

Widening Casting:

* Widening casting occurs when you convert a value from a smaller data type to a larger data type.
* It is also known as implicit casting or automatic type conversion.
* Widening casting is performed by the Java compiler automatically without the need for explicit casting operators.
* Narrowing casting occurs when you convert a value from a larger data type to a smaller data type.
* It is also known as explicit casting or manual type conversion.
* Narrowing casting requires the use of explicit casting operators, as it may result in a loss of data or precision.

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'| '\u0000' to '\uffff'

byte | 1 | 0 | -128 to 127

short | 2 | 0 | -32,768 to 32,767

int | 4 | 0 | -2,147,483,648 to 2,147,483,647

long | 8 | 0L | -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807

float | 4 | 0.0f | approximately ±1.4E-45 to ±3.4028235E+38

double | 8 | 0.0 | approximately ±4.9E-324 to ±1.7976931348623157E+308

1. Define package as used in java programming

a package is a way to organize and group related classes, interfaces, and sub-packages. It provides a hierarchical structure to organize code and avoid naming conflicts.

1. Explain the importance of using Java packages

Namespace Management: Packages act as containers for Java classes, interfaces, and other resources.

Access Control: Packages define access levels and visibility.

Modularity and Organization: Packages help organize code into logical units based on functionality, domain, or purpose.

***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 SurnameAndAge {

public static void main(String[] args) {

// Create a Scanner object to read user input

Scanner scanner = new Scanner(System.in);

// Prompt the user to enter their surname

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

String surname = scanner.nextLine();

// Prompt the user to enter their age

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

int age = scanner.nextInt();

// Calculate the number of characters in the surname

int surnameLength = surname.length();

// Print the number of characters in the surname

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

// Check if the age is even or odd and print the result

String ageType = (age % 2 == 0) ? "even" : "odd";

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

// Close the scanner

scanner.close();

}

}

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 AverageMarks {

public static void main(String[] args) {

// Create a Scanner object to read user input

Scanner scanner = new Scanner(System.in);

// Prompt the student to enter the marks for each unit

System.out.println("Enter the marks for each of the five units:");

// Variables to store the total marks and individual unit marks

double totalMarks = 0;

double[] unitMarks = new double[5];

// Loop to read the marks for each unit

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

System.out.print("Enter marks for unit " + (i + 1) + ": ");

unitMarks[i] = scanner.nextDouble();

totalMarks += unitMarks[i];

}

// Calculate the average marks

double averageMarks = totalMarks / 5;

// Display the average marks with two decimal places

System.out.printf("Average marks: %.2f%n", averageMarks);

// Close the scanner

scanner.close();

}

}

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) {

// Create a Scanner object to read user input

Scanner scanner = new Scanner(System.in);

// Prompt the user to enter an integer

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

int number = scanner.nextInt();

// Check divisibility for integers in the range of 0-9

for (int divisor = 0; divisor <= 9; divisor++) {

// Check if the number is divisible by the current divisor

if (number % divisor == 0) {

// Print the divisibility statement

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

}

}

// Close the scanner

scanner.close();

}

}

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

public class MultiplesInRange {

public static void main(String[] args) {

int start = 71;

int end = 150;

System.out.println("Multiples of 2, 3, and 7 within the range " + start + " to " + end + ":");

for (int i = start; i <= end; 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) {

// Create a Scanner object to read user input

Scanner scanner = new Scanner(System.in);

// Prompt the user to enter the first number

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

double num1 = scanner.nextDouble();

// Prompt the user to enter the operation (+, -, \*, /)

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

String operation = scanner.next();

// Prompt the user to enter the second number

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

double num2 = scanner.nextDouble();

double result = 0.0;

// Perform the selected operation

switch (operation) {

case "+":

result = num1 + num2;

break;

case "-":

result = num1 - num2;

break;

case "\*":

result = num1 \* num2;

break;

case "/":

result = num1 / num2;

break;

default:

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

// Exit the program

System.exit(0);

}

// Display the result

System.out.println("Result: " + result);

// Close the scanner

scanner.close();

}

}