**BIDWELL WARIO SCT221-0819/2022**

**JAVA ASSIGNMENT**

|  |
| --- |
|  |

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

**Definition:** Primitive data types are the most basic data types built into the language. They store actual values directly in the memory where the variable is allocated. They have a fixed size.

**Definition:** Reference data types are more complex and represent objects. They store references (or addresses) to the actual data in the memory, not the data itself. They do not have a fixed size as they can represent more complex structures.

**Define the scope of a variable**

The scope of a variable refers to the part of the program where the variable can be accessed and used. In Java, variables can have different scopes based on where they are declared. The two main types of variable scope are local and global variables.

**Local Variables**

**Definition**: Local variables are declared within a method, constructor, or block of code. (hint: local and global variable. They are only accessible within the method, constructor, or block where they are declared.

**Global Variables**

**Definition:** Global variables, also known as class-level variables, are declared outside any method, constructor, or block, but within a class. They are accessible from any method, constructor, or block within the class. If they are declared as public, they can be accessed from outside the class as well.

**Why is initialization of variables required.**

**Avoiding Undefined Behavior**: When a variable is declared but not initialized, it holds an undefined value. Using such variables can lead to unpredictable behavior and bugs in the program.

**Ensuring Predictable Results**: Initializing variables ensures that they start with a known value, making the program’s behavior more predictable and easier to debug.

**Memory Safety**: Proper initialization helps in managing memory efficiently and prevents issues like accessing invalid memory locations.

**Compiler Requirements**: Some programming languages, including Java, require variables to be initialized before they are used. This is to ensure that the program does not attempt to use variables with undefined values.

**Differentiate between static, instance and local variables.**

**Static Variables**  are variables declared with the static keyword within a class, but outside any method, constructor, or block and areaccessible throughout the class and shared among all instances of the class.

**Instance Variables** are variables declared within a class but outside any method, constructor, or block and are accessible throughout the class, but each instance of the class has its own copy.

**Local Variables** are declared within a method, constructor, or block and are accessible only within the method, constructor, or block where they are declared.

**Differentiate between widening and narrowing casting in java.**

Widening Casting is the converting of a smaller data type to a larger data type and it is done automatically by the Java compiler while Narrowing Casting is the converting of a larger data type to a smaller data type must be done manually by the programmer.

**The following table shows data type, its size, default value and the range. Filling in the**

**missing values.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Size(in bytes)** | **Default value** | **Range** |
| boolean | 1 | false | True, false |
| char | 2 | \0000 | ‘\0000’ to ‘\ffff’ |
| byte | 1 | 0 | -128 to 127 |
| short | 2 | 0 | -215 to +215-1 |
| 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 ±3.40282347E+38F (6-7 decimal digits) |
| double | 8 | 0.0d | -1.8E+308 to +1.8E+308 |

**Define class as used in OOP**

a class is a blueprint or template for creating objects**.**

**Explain the importance of classes in Java programming.**

**1. Encapsulation**

Classes encapsulate data and methods that operate on that data, providing a clear structure and protecting the internal state of objects. This encapsulation helps in maintaining data integrity and reduces the complexity of the code1.

**2. Reusability**

Once a class is defined, it can be reused to create multiple objects. This promotes code reuse and reduces redundancy, making the development process more efficient.

**3. Modularity**

Classes help in organizing code into logical units. Each class can represent a specific concept or entity, making the code more modular and easier to manage.

**4. Inheritance**

Classes support inheritance, allowing new classes to inherit attributes and methods from existing classes. This promotes code reuse and helps in creating a hierarchical structure of classes, which can be easily extended and maintained.

**5. Abstraction**

Classes provide a way to abstract real-world entities into code. By defining classes, developers can create objects that represent real-world concepts, making the code more intuitive and easier to understand.