

Task3

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

-Primitive data types are the data types provided by java

-The size is fixed

-To create a variable of primitive type 'new' keyword is not used

-Reference data types are the data types created by a programmer.

The size depends on the number of member variables and their types.

To create a variable of reference type 'new' keyword is used.

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

Scope of a variable is the block of code in the entire program where the variable is declared, used, and can be modified.

3. Why is initialization of variables required?

-To give it a correct initial value.

-To avoid errors.

4. Differentiate between static, instance and local variables.

Static variable is defined outside a method at the class level.

Instance variable is defined outside a method at class level.

Local variable is defined within a method or a code block.

5. Differentiate between widening and narrowing casting in java.

Widening conversions preserve the source value but can change its representation. This occurs if you convert from an integral type to Decimal, or from Char to String. A narrowing conversion changes a value to a data type that might not be able to hold some of the possible values.

6. 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 '\xffff'
Byte	1	0	-27 to +27-1
Short	2	0	-215 to +215-1
Int	4	0	-231 to +231-1
Long	8	0L	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
Float	4	00.0f	3.4E-38 to 3.4E+38
Double	8	0.0d	-1.8E+308 to +1.8E+308

7. Explain the importance of using Java packages

- 1) Java package is used to categorize the classes and interfaces so that they can be easily maintained.
- 2) Java package provides access protection.
- 3) Java package removes naming collision.

8. Explain three controls used when creating GUI applications in Java language.

- Label - Is used to provide a descriptive text string that cannot be changed directly by the user.
- TextField - Used to get text input from the user into the program for processing.
- Button - Used to execute blocks of code in a program when clicked by the user.
- Text area - Used to display text.

9. Explain the difference between containers and components as used in Java.

The Container class is another subclass of Component. A Container is a component that can contain other components (including other containers) while components represents visual elements of a Graphical User Interface. Its subclasses include Button, Checkbox, TextField, Choice, and Canvas.

10. Write a Java program to reverse an array having five items of type int.

```
import java.util.*;
import java.util.stream.*;
public class Main
{
    public static void main(String[] args) {
```

```

Integer[] intArray = {10,20,30,40,50,};

//print array starting from first element
System.out.println("Original Array:");
for(int i=0;i<intArray.length;i++)
    System.out.print(intArray[i] + " ");

System.out.println();

//print array starting from last element
System.out.println("Original Array printed in reverse order:");
for(int i=intArray.length-1;i>=0;i--)
    System.out.print(intArray[i] + " ");
}
}

```

11. Programs written for a graphical user interface have to deal with “events.”

Explain what is meant by the term event.

Give at least two different examples of events, and discuss how a program might respond to those events.

Event - is an object that is created when something changes within a graphical user interface.

Example 1 - When the user clicks a button, chooses a menu item, presses Enter in a text field. The result is that an actionPerformed message is sent to all action listeners that are registered on the relevant component.

Example 2 - When the user changes the value of the scrolling component, it receives an instance of AdjustmentEvent .

12. Explain the difference between the following terms as used in Java programming.

Polymorphism and encapsulation

Polymorphism allows program code to have different meanings or functions while encapsulation is the process of keeping classes private so they cannot be modified by external codes.

Method overloading and method overriding

Overloading occurs when two or more methods in the same class have the same name but different parameters whereas overriding occurs when the method signature is the same in the superclass and the child class.

Class and interface

A class can be inherited by another class using the keyword 'extends'. An Interface can be inherited by a class using the keyword 'implements' and it can be inherited by another interface using the keyword 'extends'. A class can contain constructors. An Interface cannot contain constructors.

Inheritance and polymorphism

In inheritance, we create new classes that inherit features of the superclass while polymorphism decides what form of method to execute.

Inheritance applies to classes, whereas polymorphism applies to methods.