

LYNETTE NJERI IRUNGU

21/04073

JAVA PROGRAMMING

Task3

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

- Primitive data types are already defined in Java while reference data types can be defined by the user.
- Primitive data types specify the size and type of variable values while reference data types specify the reference/address of the variable values.

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. Outside of all functions which are called global variables. Inside a function or a block which is called local variables,

3. Why is the initialization of variables required?

- It prevents bugs.
- it can prevent null reference errors down the line
- it ensures that all data objects and variables have a known starting value

4. Differentiate between static, instance and local variables.

A local variable is a variable that is declared inside the body of the method .

A static variable is a variable that is declared as static

Instance variable is a variable that is declared inside the class but outside the body of the method

5. Differentiate between widening and narrowing casting in java.

Widening casting is a type of casting that converts a low data type to high data type

While Narrowing casting is a type of casting that converts a low data type to a high data type

6. The following table soothes the ta type, its size, default values, 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	-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

- define their own packages to bundle a group of classes
- creates a new namespace so there won't be any name conflicts with names in other packages

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

- Button - Used to execute blocks of code in a program when clicked by the user.
- Checkbox - Used to display options to the user, where the user can select more than one option.
- Label - Is used to provide a descriptive text string that cannot be changed directly by the user..

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

The abstract foundation class for AWT's user interface controls without menus is the class Component. In graphical form, a component represents an object. AWT's containers belong to the class Container, which serves as their superclass. Additional AWT components may be contained by the container object

abstract foundation class for AWT's user interface controls without menus is the class Component.

In graphical form, a component represents an object.

AWT's containers belong to the class Container, which serves as their superclass.

Additional AWT components may be contained by the container object

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

```
class Shapes {
```

```

    public void area() {
        System.out.println("Find area ");
    }
    public void area(int r) {
        System.out.println("Circle area = "+3.14*r*r);
    }

    public void area(double b, double h) {
        System.out.println("Triangle area="+0.5*b*h);
    }
    public void area(int l, int b) {
        System.out.println("Rectangle area="+l*b);
    }

}

class Main {
    public static void main(String[] args) {
        Shapes myShape = new Shapes(); // Create a Shapes object

        myShape.area();
        myShape.area(5);
        myShape.area(6.0,1.2);
        myShape.area(6,2);

    }
}

```

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.

The change in the state of an object or behavior by performing actions is referred to as **an Event**

For example:, when the user clicks a button, the program can display a dialog box.

- When the user moves the cursor in a container, the cursor can change its shape.

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

Polymorphism and encapsulation

Polymorphism ensures that the proper method will be executed based on the calling object's type. Encapsulation allows you to control access to your object's state while making it easier to maintain or change your implementation at a later date.

Method overloading and method overriding

. When the method signature (name and parameters) are the same in the superclass and the child class, it's called method overriding. When two or more methods in the same class have the same name but different parameters, it's called method overloading.

Class and interface

A Class can be defined as a collection of objects that have similar types of properties. It is a logical entity that can be seen as a blueprint for creating objects. The interface in Java can be defined as the blueprint of the class.

Inheritance and polymorphism

Inheritance is one in which a new class is created that inherits the properties of the already existing class. Polymorphism is that in which we can perform a task in multiple forms or ways. It is applied to the functions or methods.

13. Using examples, explain the two possible ways of implementing polymorphism. Show your code in java.

Method overloading is the process that can create multiple methods of the same name in the same class, and all the methods work in different ways.

```
class Shapes {
    public void area() {
        System.out.println("Find area ");
    }
    public void area(int r) {
        System.out.println("Circle area = "+3.14*r*r);
    }

    public void area(double b, double h) {
        System.out.println("Triangle area="+0.5*b*h);
    }
    public void area(int l, int b) {
        System.out.println("Rectangle area="+l*b);
    }
}

class Main {
    public static void main(String[] args) {
        Shapes myShape = new Shapes(); // Create a Shapes object

        myShape.area();
    }
}
```

```
    myShape.area(5);
    myShape.area(6.0,1.2);
    myShape.area(6,2);

}
}
```

Method overriding is the process when the subclass or a child class has the same method as declared in the parent class.

Example of method overriding in Java

```
class Vehicle{
    //defining a method
    void run(){System.out.println("Vehicle is moving");}
}
//Creating a child class
class Car2 extends Vehicle{
    //defining the same method as in the parent class
    void run(){System.out.println("car is running safely");}

    public static void main(String args[]){
        Car2 obj = new Car2();//creating object
        obj.run();//calling method
    }
}
```

LYNETTE NJERI IRUNGU
21/04073