Tabby ngunjiri

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

Primitive data type-they are the most basic data type available within the java language, which are eight in number, they include: Boolean, integer, string, short, long, float and double.

Imprimitive /reference data type- this is a kind of data type created using constructor of the classes which are used to access objects. They cannot be changed.

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

A scope of a variable is the area where the variable can be accessed whether by whole program or in a block of code.

Therefore, a global variable is the variable which can be accessed by any function in a program.

Then a local variable is the variable declared within the method.

3. **Why is initialization of variables required?**

This is to prevent bug where they are initialized to prevent null errors.

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

Static variable-they are variable that are runner in the whole program

Instance variable- type of variable outside constructor, method and blocks.

Local variable- they are the variable inside block of code.

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

   Widening is the process of converting lower data type to higher data type while

   Casting is the process of converting from one data type to another. (type conversion)

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 '\ffff'
Byte		0	-2^7 to $+2^7-1$
Short		0	-2^{15} to $+2^{15}-1$
Int	4	**0**	-2^{31} to $+2^{31}-1$
Long		0L	**-10^{308} to $+10^{308}$**
Float	4	00.0f	**-10^{38} to $+10^{38}$**
Double	8	**0.0d**	-1.8E+308 to +1.8E+308

(Hint **answer is in bold)**

7. **Explain the importance of using Java packages**

✓ It is used to categorise the classes interfaces can easily be maintained.
✓ It can be easily be maintained
✓ It can be easily be protected

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

- **Java swing Button control-** to create a platform called button that perform independent implementation.
- **Java swing checkbox-** it is used to turn an option on or off
- **Java swing combo box-**this is the control that displays a pop up where a user chooses his or her choice.

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

Component-they are objects with graphical representation.
Container – it provide space where components are located.

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

```
1. public class ReverseArray {
2.     public static void main(String[] args) {
3.         //Initialize array
4.         int [] arr = new int [] {41, 32, 43,4 4,4 5};
5.         System.out.println("initial array: ");
```

```
6.          for (int i = 0; i < arr.length; i++) {
7.              System.out.print(arr[i] + " ");
8.          }
9.          System.out.println();
10.         System.out.println(" reverse  Array: ");
11.         //Loop through the array in reverse order
12.         for (int i = arr.length-1; i >= 0; i--) {
13.             System.out.print(arr[i] + " ");
14.         }
15.     }
16. }
```

11.Programs written for a graphical user interface have to deal with "events." Explain what is meant by the term event.

An event is the process of changing in the state of object behaviors by performing actions

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

Background event -this is the kind of event that requires end user interaction. this is When an operating system interupts hardware or software failure.

Foreground events-this is an event that necessitate the user's participation. Which are produce as a result of user interaction with graphical user interface.

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

- **Polymorphism and encapsulation**

Polymorphism is the instance where there are many classes relating to each other through inheritance.

Encapsulation is when the variable and methods are integrated as a single unit.

Method overloading and method overriding

Overloading-this is the form of polymorphism that uses object oriented programming.

Overriding-this occurs when su class have the same method as the parent class.

- **Class and interface**

Class is a logical template for creating object that share common properties ad method.

Interface-this is the reference type in java which is similar to a class also we can say it is a collection of abstract methods.

- **inheritance and polymorphism**
- **inheritance** is the concept that acquires the properties from one class to another classes.
- **Polymorphism-**this is the ability of a class to provide differet implementation of methods depending on the type of objects that is passed to method.

14. **using examples, explain the two possible ways of implementing polymorphism. Show your code in java.**

- **By overloading**

```
static int plusMethod(int x, int y) {
  return x + y;
}

static double plusMethod(double x, double y) {
  return x + y;
}

public static void main(String[] args) {
  int myNum1 = plusMethod(8, 5);
  double myNum2 = plusMethod(4.3, 6.26);
  System.out.println("int: " + myNum1);
  System.out.println("double: " + myNum2);
}
```

- **By ovveriding**

```
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 } }
```

Tabby ngunjiri

With relevant examples, explain the following concepts as used in Java programming.

a. Mutable classes.

Explain what is meant by mutable class

A mutable class is a type of class which its object can be changed after it is created.

Write a program that implements the concept of mutable class

b. mutable classes

```
public class IntegerPair {

    int x;

    int y;



    IntegerPair(int x, int y) {

        this.x = x;

        this.y = y;

    }

}



IntegerPair p = new IntegerPair(5, 10);

// p.x = 5, p.y = 10



p.x = 50;

// p.x = 50, p.y = 10
```

Explain what is meant by immutable class

This is the type of class that its object cant be changed once the class is created

Write a program that implements the concept of immutable class

```
public class IntegerPair {
```

```
    private final int pairx;

    private final int pairy;



    IntegerPair(int pairx, int pairy) {

        this.pairx = pairx;

        this.pairy = pairy;

    }

}



IntegerPair p = new IntegerPair(5, 10);



p.pairx = 50;

// Compilation error: cannot assign a value to a final variable
```

c. Explain the situations where mutable classes are more preferable than immutable classes when writing a Java program.

When defining our own class we can make objects immutable by making all field final ad private.

2. Explain what a String buffer class is as used in Java

It is used in to create mutable string objects

the syntax of creating an object of StringBuffer class

StringBuffer()

Explain the methods in the StringBuffer class

Append(String)-used to append or change the specified string with a new string.

Insert(string)-used to add a specified string at a specified location.

Replace(string)-used to modify a particular string from one to the specified one.

class Myoutput

```
1.          {
```

```
2.              public static void main(String args[])

3.              {

4.                  String ast = "hello i love java";

5.
  System.out.println(ast.indexOf('e')+"
  "+ast.indexOf('ast')+" "+ast.lastIndexOf('l')+"
  "+ast .lastIndexOf('v'));

6.              }}
```

There will be no output

c. Explain your answer in (2b) above.

Single quotation has been used inside the rounded brackets which will result to an eror.

when the eror is changed to using double quortation the program is expected to give the following output:**1 -1 8 15**

d. With explanation, write the output of the following program.

```
class Myoutput
  1.        {

  2.              public static void main(String args[])

  3.              {

  4.                   StringBuffer bfobj = new
    StringBuffer("Jambo");

  5.                   StringBuffer bfobj1 = new
    StringBuffer(" Kenya");

  6.                   c.append(bfobj1);

  7.                   System.out.println(bfobj);
```

```
8.            }

9.        }
```

There will be an eror. **(c.append(bfobj1));** the given variable C is not declared instead for the code to run it must be rectified.

e. With explanation, write the output of the following program.

```
class Myoutput

1.        {

2.            public static void main(String args[])

3.            {

4.               StringBuffer str1 = new
  StringBuffer("Jambo");

5.               StringBuffer str2 = str1.reverse();

6.               System.out.println(str2);

7.            }
```

```
8.        }
```

There will be no output because of the array in the code is wrongly placed the eror should be rectified to **public static void main(String[]args){**block of code**}**

f. With explanation, write the output of the following program.

```
class Myoutput
1.        {
2.           class output
3.           {
4.             public static void main(String args[])
5.             {
6.                char c[]={'A', '1', 'b' ,' ' ,'a' ,
  '0'};
7.                for (int i = 0; i < 5; ++i)
8.                {
9.                       i++;
10.                      if(Character.isDigit(c[i]))
11.
  System.out.println(c[i]+" is a digit");
12.
  if(Character.isWhitespace(c[i]))
13.
  System.out.println(c[i]+" is a Whitespace
  character");
```

```
14.
  if(Character.isUpperCase(c[i]))

15.
  System.out.println(c[i]+" is an Upper case
  Letter");

16.
  if(Character.isLowerCase(c[i]))

17.
  System.out.println(c[i]+" is a lower case Letter");

18.                        i++;

19.                }

20.            }

21.        }
```

the code will not run . in the output lines there are no opening double quotes hence the code wont run . by adding quotation inside therounded