1.explain the difference between primitive and reference data type

Primitive variable store the actual value while references variables store the address of the object they refer to

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

3.why is initialization of variable required

To give it a correct initial value

4.differentiate between static instance and local variable

Static variable can be accessed using the names of the class while, local variable can be accessed using the object of the demo class and instance variable are declared in the class but outside a method

5.differentiate between widening and narrowing casting in java

Widening conversation preserves the source value but can change its presentation while narrowing conserves changes a value to a data type that might not be able to hold some of the possible values

6

7.explain the importance of using java packages

To avoid name conflict

To write a better maintainable code

8.explain three controls used when creating GUL applications in java languages

1. In the Projects window, right-click the NumberAddition node and choose New > Other .
2. In the New File dialog box, choose the Swing GUI Forms category and the JFrame Form file type. Click Next.
3. Enter NumberAdditionUI as the class name.
4. Enter my. numberaddition as the package

5.click finish

9.explain the difference between container and component as used in java

Java's Component class represents visual elements of a Graphical User Interface. . The Container class is another subclass of Component. **While a Container is a component that can contain other components (including other containers)**.

10.write a java reverse an array having five items to type int

1. Using Temp array.
2. Using Swapping.
3. Using Collections. reverse() method.
4. Using StringBuilder. append() method
5. Programs written for a graphical user interface have to deal with “events.”

Explain what is meant by the term event.

**Changing the state of an object or behavior by performing actions** give at least two different examples of events, and discuss how a program might respond to those events

1.The user clicks a mouse button.

2.The user presses a key on the keyboard .

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

Polymorphism and encapsulation

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

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 overriding. When two or more methods in the same class have the same name but different parameters, it's called overloading.

class and interface

An interface is a reference type in Java. It is similar to class. It is a collection of abstract methods. A class implements an interface, thereby inheriting the abstract methods of the interface.

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.

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

class Shapes{

public void area( ) {

System.out.println

("The formula for area of ");

}

}

class Triangle extends Shapes {

public void area() {

System.out.println("Triangle is ½ \* base \* height ");

}

}

class Circle extends Shapes {

public void area() {

System.out.println("Circle is 3.14 \* radius \* radius ");

}

}

class Main {

public static void main(String[] args) {

Shapes myShape = new Shapes(); // Create a Shapes object

Shapes myTriangle = new Triangle(); // Create a Triangle object

Shapes myCircle = new Circle(); // Create a Circle object

myShape.area();

myTriangle.area();

myShape.area();

myCircle.area();

}

}

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

a. Mutable classes.

Explain what is meant by mutable class

**A mutable object can be changed after it's created**, and an immutable object can't

Write a program that implements the concept of mutable class

onsider the below example of the mutable class:

1. **public** **class** JtpExample {
2. **private** String s;
3. JtpExample(String s) {
4. **this**.s = s;
5. }
6. **public** String getName() {
7. **return** s;
8. }
9. **public** **void** setName(String coursename) {
10. **this**.s = coursename;
11. }
12. **public** **static** **void** main(String[] args) {
13. JtpExample obj = **new** JtpExample("JavaTpoint");
14. System.out.println(obj.getName());
15. **// Here, we can update the name using the setName method.**
16. obj.setName("Java Training");
17. System.out.println(obj.getName());
18. }
19. }

b. Immutable classes.

Explain what is meant by immutable class

Immutable class in java means that **once an object is created, we cannot change its content**

Write a program that implements the concept of immutable class

|  |
| --- |
|  |

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

|  |  |
| --- | --- |
| Mutable | Immutable |
| Fields can be changed after the object creation | Fields cannot be changed after object creation |
| Generally provides a method to modify the field value | Does not have any method to modify the field value |
| Has Getter and Setter methods | Has only Getter method |
| Example: StringBuilder, java.util.Date | Example: String, Boxed primitive objects like Integer, Long and etc |

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

the syntax of creating an object of StringBuffer class

Explain the methods in the StringBuffer class

b. Write the output of the following program.

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. }
7. }

Ans 114815

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

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. }

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. }

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. }