|  |  |  |
| --- | --- | --- |
| **Q.21** | Discuss all points about key word “this” and “super” with examples (while writing constructors) | |
| **Ans:** | **SUPER**  It is a reference variable used to refer the immediate parent classobject  - super() invokes immediate parent class constructor  - Call member (variables & methods) of parent class  **Syntax :** super.baseclassMemberName  **Example:**  class Base{  public void baseMethod() {  System.out.println(“Base);  }  }  Class Derived extends Base{  Public void derivedMethod() {  Super.baseMethod();  System.out.println(“Derived);  }  }  Class Test {  Public static void main(String args[]) {  Derived derived = new new Derived();  Derived.derivedMethod();  }  }    **THIS**  this keyword:  this keyword is used to refer the current object. As shown in the below example, the constructor parameters are shadowing the instance variables. Therefore we can use this keyword to make difference between the local variables/parameters and instance variables. | |
| **Q.22** | How will you write varargs (what conditions must be followed). | |
| **Ans:** | Varargs(Variable Argument List) or ellipsis(…)  //Valid Code  Void print(int a,int b,String…c)  { //code }  //Invalid Code  Void print(int a,int b…,float c)  { //code }  Rule: varargs can be used only in the final argument position. | |
| **Q.23** | Explain all points about :   1. Final variable 2. Final method 3. Final class | |
| **Ans:** | **Final variable:**   * Behaves like a constant; i.e. once initialized, its value cannot be changed. * Example: final int i=10;   **Final Method:**   * Method declared as final cannot be overridden in subclasses * Its values cannot change once initialized * Examples: A class or method cannot be abstract & final at the same time.   **class A {**  **Public final int add(int a, int b)**  **{ return a+b ;}**  **}**  **Final class:**   * Cannot be sub-classed at all * Examples: String and StringBuffer class | |
| **Q.24** | Difference between final and abstract class | |
| **Ans:** | **Final Classes**   * final classes are the way we can prevent class being extended * We can instantiate final class and immutable objects can be created * These cannot contain abstract methods * Must have all the method implementations in it * Eg: String class | A**bstract Classes**   * Abstract classes are always extended, for situation like no abstract methods, these still can have member elements that makes body and which can be inherited. * Cannot be instantiated. So, cannot create immutable objects * E.g.: HTTP Servlet class |
| **Q.25** | By default interface data members are \_\_\_\_\_\_\_\_\_\_\_ | |
| **Ans:** | By default interface data members are **PUBLIC, STATIC and FINAL**. | |
| **Q.26** | Three classes in regex package | |
| **Ans:** | The java.util.regex package primarily consists of the following three classes:   * Pattern * Matcher * PatternSyntaxException | |
| **Q.27** | Examples on pattern matching | |
| **Ans:** | Example 1:  public class RegExpTest {  public static void main(String[] argss) {  String inputStr = “Test String”;  String pattern = “Test String”;  Boolean patternMatched =  Pattern.matches(pattern,inputStr);  System.out.println(pattern.Matched);  }  }  Example 2:  package com.capgemini.lesson8;  import java.util.regex.\*;  public class RegExpTest {  public static void main(String[] args)  {  String inputStr = "Test String";  String pattern = "Test String";  boolean patternMatched = Pattern.matches(pattern, inputStr);  System.out.println(patternMatched);  /\*  \* Pattern pattern1 = Pattern.compile(","); String[] str =  \* pattern1.split("Shop,Mop,Hopping,Chopping"); for (String st : str) {  \* System.out.println(st); }  \*/  String input = "Shop,Mop,Hopping,Chopping";  Pattern pattern1 = Pattern.compile("hop");  Matcher matcher = pattern1.matcher(input);  System.out.println(matcher.matches());  while (matcher.find())  {  System.out.println(matcher.group() + ": " + matcher.start() + ": "  + matcher.end());  }  }  } | |
| **Q.28** | List and discuss ALL the Checked exception and UnChecked exception | |
| **Ans.** | * Checked Exceptions : SQLException, IOException, ClassNotFoundException * UnChecked Exceptions : NullPointerException, ArithmeticException, ArrayIndexOutOfBoundException, NumberFormatException | |