|  |  |  |
| --- | --- | --- |
| **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 class  object  - super() invokes immediate parent class constructor  - Call member (variables & methods) of parent class  **Syntax :** super.baseclassMemberName    **THIS**  this keyword:  this keyword is used to refer the current object. As shown in the above 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:** |  | |
| **Q.23** | Explain all points about :   1. Final variable 2. Final method 3. Final class | |
| **Ans:** |  | |
| **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. * Can not be instantiated. So, cannot create immutable objects * Eg: HTTP Servlet class |
| **Q.25** | By default interface data members are \_\_\_\_\_\_\_\_\_\_\_ | |
| **Ans:** | By default interface data members are **FINAL**. | |
| **Q.26** | Three classes in regex package | |
| **Ans:** |  | |
| **Q.27** | Examples on pattern matching | |
| **Ans:** | Example 1:  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 | |