Capgemini Interview Questions:

1. Sort HashMap by values?

scores.put("David", 95);

scores.put("Jane", 80);

scores.put("Mary", 97);

scores.put("Lisa", 78);

scores.put("Dino", 65);

Sort by Values?

Answer:

/\* Fetch all hash map values into entry set \*/

Map.Entry<String,Integer> allScoreEntry = scores.entrySet();

/\* declare arraylist to add list of entries into hashmap\*/

List<Map.Entry<String,Integer>> listOfEntries = new ArrayList<Map.Entry<String,Integer>>();

/\* Iterating HashMap One by one and adding it into ArrayList/\*

for(Map.Entry<String,Integer> entry : allScoreEntry )

listOfEntries.add(entry);

/\* Sort the list of hashmap entries using comparator /\*

Collections.sort(listOfEntries, new Comparator<Map.Entry<String,Integer>>()) {

Public int compare(Map.Entry<String,Integer> o,Map.Entry<String,Integer> o1) {

return 0.getValue() – 01.getValue();

}

};

/\*declare Linkedhasmap to hold list of hashmap entries/\*

LinkedHasMap<String,Integer> allEntries = new LinkedHashMap< String,Integer >();

/\* add all entries into linked hash map by iterating arraylist/\*

for(Map.Entry<String,Integer> o : listOfEntries) {

allEntries.add(o.getKey(),o.getValue());

}

/\* Print all Linked HasMap entries\*/

for(Map.Entry<String,Integer> o : listOfEntries) {

Sysout(“key:”+o.getKey()+”and value is:” + o.getValue());

}

(2) Result:

class Base {

protected void foo() {

}

}

class Derived extends Base {

void foo() {

}

}

public class Main {

public static void main(String args[]) {

Derived d = new Derived();

d.foo(); }

}

Answer:

Compile time error: Cannot reduce the visibility of a method in child class: week Access modifier

(3) Difference between System classes and Runtime classes in java?

Runtime Class: it is a class of Lang package, which is meant for retrieving the memory details of the system during runtime of the programme, this is also meant for enabling forced garbage collection, and it provides the method for running a process during runtime of the programme.

Code:

Runtime rt = Runtime.getRunTime ();

//rt.freeMemeory()---->Available Memory

//rt.totalmemory()----->Total Memory

//rt.exec(“chrome.exe”)------>To launch an application

System class: It is a class of Lang package, which is meant for providing the constants for pointing to console either for taking the input or for sending the output to the console.

Example:

System.out.println (“Valid message”);

System.err.print (“Error message”);

Int a = System.in.read ();

If ((char) a == ‘e’)

System. Exit (0);

else

System.out.println (“Normal Ending”);

(4)What is is-a and has-a relationship in java?

(5)Difference between coupling and cohesion in java?

Coupling:

Coupling indicates level of dependency between classes, in a good software design classes should be loosely coupled to each other

Refers to connection / dependency between two classes. Means, changing in one class should not have major change in other classes.

Example: spring framework, Interfaces.

To Do: what is the need to for spring framework when your interface already providing the

Loose coupling.

Cohesion:

Cohesion indicate level of logical connection of methods and attributes used in a class

In a good software design, classes should be highly cohesive.

Refers to what a class does. In other terms, it refers its work. A class with high cohesion is focused on what it does its specific to what it does

Example:

A class "Email Utility" sends email, checks for pending email to be sent. If it has operations related to customer or any other entity, its low cohesion. Another Example, "Ticketmaster" class. Which creates a new ticket, updates ticket, gets all tickets, gets ticket by ID, and deletes ticket. Etc. Therefore, it is doing operations specific to a ticket. It does not update customer or does not send an email

(6)Diff between doget () and dopost () in java?

(a)HttpServlet class extends the generic servlet.

(b) It is commonly used when developing servlets that receive and process requests.

(c) The http servlet class provides specialized methods that handle the various types of http requests.

(d)Methods are: doGet(),doOptions(),doDelete(),doHead(),doPost(),doPut(),dpTrace().

A Servlet developer typically overrides one of these methods.

Example: https://www.youtube.com/watch?v=cWo1UjJ5\_kQ (Refer for coding)

**Handling HTTP Get Requests:**

1. You must override doGet () Method.
2. The doGet () Method receives HttpServlet request and HttpServletResponse object and must throw java.io.ServletException and java.io.IOException.

Example:

Public void doGet (HttpServeletReq req, HttpServletResponse res) throws ServletException, IOException {

res.setContenetType(“text/html”);

PrintWrite out = res.getWriter();

Out.println(“vvvvvvv”);

}

**Handling HTTP POST Requests:**

The doPost () method is overridden to process any http post request that are sent to the servlet.

(a) It uses get Parameter() method of httpServletRequest

(b) The doPost() method receives HttpServlet request and HttpServletResponse object and must throw must throw java.io.ServletException and java.io.IOException.

Example:

Public void doPost (HttpServeletReq req, HttpServletResponse res) throws ServletException,IOException {

res.setContenetType(“text/html”);

PrintWrite out = res.getWriter();

Out.println(“vvvvvvv”);

}

doGet () doPost ()

(a)It is default method of any http request (a) it is not default.

(b)It is designed to gather data from server by generating req. (b) It is designed to send however, we can send unlimited amount of data. Unlimited amount of data to

Server along with req.

(c)The form page generated query string is visible in the browser (c) The form page generated query string is not visible in the browser

(d)Not suitable for encrypt operations (d) Suitable for encrypt operations.

(e) Not suitable for file uploading operation (e) suitable for file Uploading operation.

(f) It also supports sending multiple part form data whereas doGet does not.

(7) Difference between static synchronized and non-static synchronized in java?

(8)What is treeMap?

(9)Data hiding in java?

(10)What is serialization and what is the purpose serialVersionUID and How to make class serializable?

(11)Can we initialize the final variable in constructor?

(12) Servlet redirect vs forward.

(13)How to do Spring security in spring boot?

(14) What is no class deffound error?

**Accenture**

(1) Java8 new features?

(2)What are the intermediate and terminal operations? (Examples)

(3)Can we override static methods in java?

(4) Can we implement functional interface like other interfaces?

(5)What is the use of @Functional interface and is it mandatory?

(6) Can we throw normal checked exception using throw keyword?

**(7) try {**

**int a = 10/0;**

**}**

**catch(IllegalStatement Exception) {**

**}**

**finally {**

**sysout("")**

**}**

(8)Difference between @controller and @RestController

(9)Diff between traditional dependency injection and spring dependency injection?

**(10)**

**try {**

**int a = 10/0;**

**}**

**catch (Exception) {**

**}**

**finally {**

**return 10;**

**}**

**return 20;**

(11)What is the unreachable code?

(12)What is @Qualifier annotation?

(13) default methods in java8?

(14) Underlying data structure of arraylist.

(15) Hashmap internal implementation?

(16) Contract between hashcode and equals?

(17)Having a list of numbers and how you get size of even numbers using java 7 and java8?

(18)What is the changes that made to hashmap in java8?

(19) class Employee {

private static final int a;

}

Does the above code compile?

(20)Can we override static methods in java?

(21) Immutable class in java?

(22)Jvm Architecture?

(23)What is metaspace?

(24)What is actuator?

(25)Spring mvc flow and what is role of view resolver?

(26)Syntax for optional?