

Objective

1. B. ArithmeticException
2. B. start exception finally end
3. A. car maruti tata
4. A. MyException
5. A. Parent p = **new** Parent(); Parent.Child c = p.**new** Child();
6. B. Runtime error
7. D. Range
8. A.
9. A. jspinit, jspservice, jspdestroy
10. C. mouseover

Subjective

Answer 1:

ArrayList	LinkedList
An ArrayList uses dynamic array to store the elements.	A LinkedList uses a doubly linked list to store the elements.
Manipulation in ArrayList is slow because it uses an array. Removal of elements in the array leads to shifting of bits in memory.	Unlike ArrayList in LinkedList it's easier to manipulate data because there is not shifting of bits in memory.
An ArrayList class can only act as a list because it implements List interface.	LinkedList class can act as both list and queue, because it implements List and Deque interfaces.
ArrayList is better for storing and accessing data.	LinkedList is better for manipulating data.

Example:

```
import java.util.ArrayList;  
  
import java.util.LinkedList;  
  
import java.util.List;
```

```

class MyClass{
    public static void main(String args[]){

        List<String> a=new ArrayList<String>();
        a.add("Varun");
        a.add("Raj");

        List<String> b=new LinkedList<String>();
        b.add("Sam");
        b.add("Cook");

        System.out.println("arraylist: "+a);
        System.out.println("linkedList: "+b);
    }
}

```

Answer 2:

JSP has a built-in mechanism for error handling which is a special page that can be used to handle every errors in the web application. There may be exceptions which occur in a web application it is necessary to handle the them.

One of the ways to do it using the page directive

Here we have to define and create a page to handle the exceptions i.e the error.jsp page. Wherever there might be a chance of an error occurring the errorPage attribute is defined. Create an error.jsp as shown below:

```

error.jsp

<%@ page isErrorPage="true" %>

<h3>Sorry an exception occurred!</h3>

Exception is: <%= exception %>

```

Then use `<%@ page errorPage="error.jsp" %>` on top of the pages where there is a possibility for an exception.

Answer 3:

There are different types of tags available in JSP. Few of them are mentioned below:

1. JSP Declaration Tags
2. JSP Expression Tags
3. JSP Directive Tags
4. JSP Scriptlet Tags
5. JSP Flow Control Tags
6. JSP Action Tags
7. JSP Comment Tags

Answer 4:

An http is known as a stateless protocol because each and every request is executed independently without any knowledge of the requests that were executed prior to it i.e once the transaction ends the connection between the browser and the server is also lost.

Answer 5:

There are many ways to iterate over a list

1. Simple For loop
2. Enhanced For loop
3. Iterator
4. ListIterator
5. While loop
6. `Iterable.forEach()` util
7. `Stream.forEach()` util

The Fail Fast system is a system that shuts down immediately after an error is reported. All the operations will be aborted instantly in it.

There are different ways to avoid Fail Fast

1. Using of loop to avoid `ConcurrentModificationException`
2. We can use the `remove()` method of Iterator to remove the object from the underlying collection object.

3. We can iterate over the array instead of iterating over the collection class. Following this way, we can work very well with small-sized lists, but this will degrade the performance if the array size is very large.
4. Locking the list by putting it in the synchronized block is another way to avoid the concurrent modification exception. This is not an effective approach; it is not using the sole purpose of multi-threading.
5. The classes `ConcurrentHashMap` and `CopyOnWriteArrayList` of JDK 1.5 or higher versions can also help us in avoiding concurrent modification exceptions.

Answer 6:

`getSession()` : Returns the current session associated with this request, or if the request does not have a session, creates one.

`getSession(true)` : Returns the current `HttpSession` associated with this request, if there is no current session, returns a new session

`getSession(false)` : Returns the current `HttpSession` associated with this request, if there is no current session, returns null.

In security scenario where we always needed a new session, we should use `request.getSession(true)`.

Answer 7:

A memory leak is a situation where unused objects occupy unnecessary space in memory. Unused objects are typically removed by the Java Garbage Collector (GC) but in cases where objects are still being referenced, they are not eligible to be removed. As a result, these unused objects are unnecessarily maintained in memory.

Memory leaks block access to resources and cause an application to consume more memory over time, leading to degrading system performance. If memory leaks are not handled, an application can eventually run out of memory and terminate with an `'OutOfMemoryError'`, which is irrecoverable.

This can be prevented by following these steps:

1. Do not create unnecessary objects.
2. Avoid String Concatenation.
3. Use String Builder.
4. Do not store a massive amount of data in the session.

5. Time out the session when no longer used.
6. Do not use the System.gc() method.
7. Avoid the use of static objects. Because they live for the entire life of the application, by default. So, it is better to set the reference to null, explicitly.
8. Always close the ResultSet, Statements, and Connection objects in the finally block.

Answer 8:

Yes, we can declare a class with the final keyword. It is called a final class. A final class cannot be extended (inherited).

Answer 9:

Using synchronized keyword, we can ensure that only one thread can use shared resource at a time and others can get control of the resource only once it has become free from the other one using it.

Answer 10:

Object cloning in Java is the process of creating an exact copy of the original object. In other words, it is a way of creating a new object by copying all the data and attributes from the original object. This is only possible by implementing clone() method of the java.lang.Object class. The clone() method saves the extra processing task for creating the exact copy of an object. If we perform it by using the new keyword, it will take a lot of processing time to be performed that is why we use object cloning. The java.lang.Cloneable interface must be implemented by the class whose object clone we want to create.

Programs

Solution for 1st Program:

```
public class StringCheck {  
    static int MAX = 256;
```

```

static boolean canMakeStr2(String str1, String str2)
{
    int[] count = new int[MAX];
    char []str3 = str1.toCharArray();
    for (int i = 0; i < str3.length; i++)
        count[str3[i]]++;

    char []str4 = str2.toCharArray();
    for (int i = 0; i < str4.length; i++) {
        if (count[str4[i]] == 0)
            return false;
        count[str4[i]]--;
    }
    return true;
}

static public void main(String []args)
{
    String str1 = "helloworld";
    String str2 = "wohe";
    if (canMakeStr2(str1, str2))
        System.out.println("true (we can form string 2 using data from string
1)");
    else
        System.out.println("false (we cannot form string 2 using data from
string 1)");
}

```

```
}
```

Solution for 2nd Program:

```
import java.util.HashMap;
```

```
public class ArraySum {
```

```
    public static void printpairs(int arr[], int X) {
```

```
        HashMap<Integer, Integer> elementIndexMap = new  
HashMap<Integer, Integer>();
```

```
        for (int i = 0; i < arr.length; i++) {
```

```
            elementIndexMap.put(arr[i], i);
```

```
        }
```

```
        int ref = 0;
```

```
        for (int i = 0; i < arr.length; i++) {
```

```
            if (elementIndexMap.get(X - arr[i]) != null &&  
elementIndexMap.get(X - arr[i]) != i) //
```

```
            {
```

```
                System.out.println "{" + arr[i] + "," + (X - arr[i]) +  
"}");
```

```
                ref++;
```

```
            }
```

```
        }
```

```
        if (ref == 0) {
```

```
            System.out.println("Cannot get values for given sum");
```

```
        }
```

```
    }
```

```
public static void main(String[] args) {  
    int a[] = { 1, 2, 5, 3, 4 };  
    int sum = 7;  
    printpairs(a, sum);  
}  
}
```