1. ) what is metaspace and heap memory?

Metaspace : is a native memory manager in the hotspot. It is used **to manage memory for class metadata**. Class metadata are allocated when classes are loaded.

Heap memory : is **a part of memory allocated to JVM**, which is shared by all executing threads in the application. It is the part of JVM in which all class instances and are allocated. It is created on the Start-up process of JVM. It does not need to be contiguous, and its size can be static or dynamic.

2)

package com.pack;

import java.util.Scanner;

public class TableOf2 {

public static void main(String[] args) {

Scanner scan=new Scanner(System.***in***);

    System.***out***.println("enter the number");

    int num=scan.nextInt();

    int n = 1;

    while (n<=10)

    {

        System.***out***.println(num\*n);

        n++;

    }

}

}

1. check if two strings are equal or not

package com.pack;

import java.util.Scanner;

public class EqualStrings {

public static void main(String[] args) {

Scanner sc1 = new Scanner(System.***in***);

System.***out***.println("Enter String 1:");

String str1= sc1.nextLine();

Scanner sc2 = new Scanner(System.***in***);

System.***out***.println("Enter String 2:");

String str2= sc2.nextLine();

System.***out***.println( str1.equals(str2) ); //prints false

}

}

4)package com.lambda;

import java.io.\*;

import java.util.\*;

class CountChar {

static void characterCount(String inputString)

{

HashMap<Character, Integer> charCountMap

= new HashMap<Character, Integer>();

char[] strArray = inputString.toCharArray();

for (char c : strArray) {

if (charCountMap.containsKey(c)) {

charCountMap.put(c, charCountMap.get(c) + 1);

}

else {

charCountMap.put(c, 1);

}

}

for (Map.Entry entry : charCountMap.entrySet()) {

System.out.println(entry.getKey() + " " + entry.getValue());

}

}

public static void main(String[] args)

{

String str = "helloworld";

characterCount(str);

}

}

1. ) why java is platform independent

Java is platform-independent because it does not depend on any type of platform.Programs are compiled into byte code and that byte code is platform-independent. ... Any machine to execute the byte code needs the Java Virtual Machine.Hence java is a platform independent language.

1. ) can we create class as final?

Yes we can create class as final class.Final class is **to prevent the class from being subclassed**. If a class is marked as final then no class can inherit any feature from the final class. You cannot extend a final class. If you try it gives you a compile time error.

1. ) considder we have employee class with empid, empname and salary and list of employees get the the highest salary paid employee data.
2. )consider a list of duplicate values remove duplicate value and get unique values from the list

package com.pack;

import java.util.ArrayList;

import java.util.Arrays;

public class DuplicateValues {

public static <T> ArrayList<T> removeDuplicates(ArrayList<T> list)

    {

ArrayList<T> newList = new ArrayList<T>();

        for (T element : list)

        {

        if (!newList.contains(element))

        {

                newList.add(element);

            }

        }

        return newList;

    }

  public static void main(String args[])

    {

    ArrayList<Integer>

            list = new ArrayList<>(Arrays .*asList*(1, 10, 1, 2, 2, 3, 3, 10, 3, 4, 5, 5,8,8));

    System.***out***.println("ArrayList with duplicates: " + list);

    ArrayList<Integer>

    newList = *removeDuplicates*(list);

    System.***out***.println("ArrayList with duplicates removed: "  + newList);

    }

}

1. can we write try and finally without catch block what is the use

Yes, we can have try without catch block by using **finally block**. Finally block always executes even if you have exception or return statement in try block except in case of System.

}