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.

package com.pack;

import java.util.Arrays;

import java.util.List;

public class Employee {

private long id;

private String name;

private int salary;

public Employee(long id, String name, int salary) {

this.id = id;

this.name = name;

this.salary = salary;

}

public int getSalary() {

return salary;

}

public void setSalary(int salary) {

this.salary = salary;

}

@Override

public String toString() {

return "Employee [id=" + id + "," + " name=" + name + "," + " salary=" + salary + "]";

}

}

class FindEmployee {

public static void main(String[] args) {

List<Employee> employees = Arrays.asList(new Employee(101, "Jay", 5000), new Employee(109, "Atul", 3000),

new Employee(111, "Sourav", 4400));

int maxSalary = employees.stream().map(Employee::getSalary).max(Integer::compare).get();

System.out.println("Max salary of the employee:" + maxSalary);

System.out.print("Employee details:");

employees.stream().filter(emp -> emp.getSalary() == maxSalary).forEach(System.out::println);

}

}

1. )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.

1. )

package l.pack;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

import java.util.stream.Collectors;

public class StudentD {

public static void main(String[] args) {

List<Employee> empList = new ArrayList<>();

List<Employee> dlst = empList.stream().distinct().collect(Collectors.toList());

System.out.println("Welcome to College Management");

char choice;

String result;

int count = 0;

do {

if (count > 0) {

System.out.println("C : Check Student Result");

}

System.out.println("A : Add Student Result");

System.out.println("X : Exit");

Scanner sc = new Scanner(System.in);

String c = sc.next();

switch (c.toUpperCase()) {

case "A":

System.out.println("Enter student id");

int id = sc.nextInt();

System.out.println("Enter student Name");

String name = sc.next();

System.out.println("Enter marks in hindi");

int hindi = sc.nextInt();

System.out.println("Enter marks in english");

int eng = sc.nextInt();

System.out.println("Enter marks in maths");

int maths = sc.nextInt();

System.out.println("Enter marks in science");

int science = sc.nextInt();

System.out.println("Enter marks in social");

int social = sc.nextInt();

int total = hindi+eng+maths+science+social;

float per = (float) total / 5;

per = per \* 100;

if (per > 50) {

result="Pass";

} else {

result="Fail";

}

dlst.add(new Employee(id, name,hindi,eng,maths,science,social,result,total,per));

System.out.println("Student added successfully");

count++;

break;

case "C":

System.out.println("Enter Student Id to Check result: ");

int in=sc.nextInt();

for(Employee e:dlst)

{

if(e.getEmpId()==(in))

{

dlst.stream().filter(emp->emp.getEmpId()==in).forEach(System.out::println);

System.exit(0);

}

}

System.out.println("No such data found");

break;

case "X":

System.exit(1);

default:

System.out.println("Invalid choice");

break;

}

System.out.println("Do you want to continue (Y/N)");

choice = sc.next().charAt(0);

} while (choice == 'Y' || choice == 'y');

System.out.println("Bye");

System.exit(1);

}

@Override

public String toString() {

return "MyClass []";

}

}

class Employee {

private int empId;

private String empNm;

private int m1, m2, m3, m4, m5;

private String r;

private float per;

private int t;

Scanner sc = new Scanner(System.in);

public Employee() {

}

public Employee(int empId, String empNm, int m1, int m2, int m3, int m4, int m5, String r,int t, float per ) {

super();

this.empId = empId;

this.empNm = empNm;

this.m1 = m1;

this.m2 = m2;

this.m3 = m3;

this.m4 = m4;

this.m5 = m5;

this.per=per;

this.r=r;

this.t=t;

}

public int getEmpId() {

return empId;

}

public void setEmpId(int empId) {

this.empId = empId;

}

public String getEmpNm() {

return empNm;

}

public void setEmpNm(String empNm) {

this.empNm = empNm;

}

public int getm1() {

return m1;

}

public void setm1(int m1) {

this.m1 = m1;

}

public int getm2() {

return m2;

}

public void setm2(int m2) {

this.m2 = m2;

}

public int getm3() {

return m3;

}

public void setm3(int m3) {

this.m3 = m3;

}

public int getm4() {

return m4;

}

public void setm4(int m4) {

this.m4 = m4;

}

public int getm5() {

return m5;

}

public void setm5(int m5) {

this.m5 = m5;

}

public String getR() {

return r;

}

public void setR(String r) {

this.r = r;

}

public float getPer() {

return per;

}

public void setPer(float per) {

this.per = per;

}

public int getT() {

return t;

}

public void setT(int t) {

this.t = t;

}

@Override

public String toString() {

return "Student Result {Id=" + empId + ", name=" + empNm + ",marks=Subject Marks{ Hindi=" + m1 + ", English=" + m2 + ", maths=" + m3 + ", Science="

+ m4 + ", Social=" + m5 + "}, result=" + r +", total=" + t + ", percentage=" + per + "}";

}

}

}