**Lab :  
25th Apr, 2024  
Question 1:**  
Create a class Employee with attributes id, name, and salary. Implement an ArrayList of Employee objects. Write a method to sort the employees by salary in descending order and print the sorted list.  
Hint: Override compare()  
 **Question 2:   
Scenario 1: Managing a List of Students**  
You are building a simple school management system. You need to store and manage a list of students. Each student has a name and an ID.  
How would you create a Vector to store the names of the students?  
Think about how you would declare and initialize a Vector in Java.  
How would you add the name "Alice" to the Vector?  
Look into methods that allow you to add elements to a Vector.  
How would you retrieve the name of the student at index 2 in the Vector?  
Consider methods that allow you to access elements by their index.  
How would you remove the student named "Bob" from the Vector?  
Think about methods that let you remove elements either by value or index.  
How would you check if a student named "Charlie" is in the Vector?  
Look for methods that help you check for the presence of an element in the Vector.  
  
**Scenario 2: Storing Book Titles in a Library**  
You are developing a library system where you need to keep track of book titles.  
How would you create a Vector to store the titles of the books?  
Remember how to declare and initialize a Vector in Java.  
How would you add the title "Java Programming" to the Vector?  
Think about methods that allow you to add elements to a Vector.  
How would you find out how many books are currently in the Vector?  
Look into methods that provide the size or length of the Vector.  
How would you replace the book title at index 3 with "Advanced Java"?  
Consider methods that allow you to update elements in a Vector.  
How would you remove all the book titles from the Vector?  
Think about methods that clear all elements from a Vector.  
  
**Scenario 3: Tracking Inventory in a Store**  
You are tasked with managing the inventory of a store. Each item in the store has a name and a quantity.  
How would you create a Vector to store the quantities of the items?  
Think about how to declare and initialize a Vector to hold integer values in Java.  
How would you add the quantity 50 for a new item to the Vector?  
Look into methods that allow you to add elements to a Vector.  
How would you increase the quantity of the item at index 1 by 20?  
Think about how to access and update the element at a specific index.  
How would you find the index of the item with quantity 100?  
Look for methods that help you find the index of a specific element in a Vector.  
How would you check if the Vector is empty?  
  
Consider methods that let you check if a Vector has any elements or not.  
-------------------------------------------------------------------------------------------------------------------------------------  
**Scenario 1: Grocery Shopping List**  
You are creating an application to manage a grocery shopping list.  
How would you create an ArrayList to store the names of grocery items?  
Think about how you would declare and initialize an ArrayList in Java.  
How would you add the item "Milk" to the ArrayList?  
Look into methods that allow you to add elements to an ArrayList.  
How would you find out how many items are currently in the ArrayList?  
Consider methods that provide the size or length of the ArrayList.  
How would you remove the item "Eggs" from the ArrayList?  
Think about methods that let you remove elements either by value or index.  
How would you replace the item at index 2 with "Bread"?  
Look for methods that allow you to update elements in an ArrayList.  
  
**Scenario 2: Student Grades Management**  
You are developing a system to manage student grades for a class.  
How would you create an ArrayList to store the grades (as integers) of the students?  
Remember how to declare and initialize an ArrayList to hold integer values in Java.  
How would you add a grade of 85 to the ArrayList?  
Look into methods that allow you to add elements to an ArrayList.  
How would you calculate the average grade of the students in the ArrayList?  
Think about how you can sum all the grades and divide by the number of grades.  
How would you find the highest grade in the ArrayList?  
Consider iterating through the ArrayList to find the maximum value.  
How would you remove all failing grades (less than 60) from the ArrayList?  
Think about how you can iterate through the ArrayList and remove elements that meet a condition.  
  
**Scenario 3: Library Book Titles**  
You are developing a library system to keep track of book titles.  
How would you create an ArrayList to store the titles of the books?  
Think about how you would declare and initialize an ArrayList in Java.  
How would you add the title "The Great Gatsby" to the ArrayList?  
Look into methods that allow you to add elements to an ArrayList.  
How would you check if the title "Moby Dick" is in the ArrayList?  
Consider methods that help you check for the presence of an element in the ArrayList.  
How would you remove all book titles from the ArrayList?  
Look for methods that clear all elements from an ArrayList.  
How would you sort the book titles in alphabetical order?  
Think about methods or utilities in Java that can help you sort an ArrayList.  
  
**Scenario 4: Employee Names in a Company**  
You are tasked with managing the names of employees in a company.  
How would you create an ArrayList to store the names of employees?  
Remember how to declare and initialize an ArrayList in Java.  
How would you add the employee name "John Doe" to the ArrayList?  
Look into methods that allow you to add elements to an ArrayList.  
How would you find the index of the employee named "Jane Smith"?  
Consider methods that help you find the index of a specific element in an ArrayList.  
How would you replace the employee name at index 1 with "Emily Johnson"?  
Think about methods that allow you to update elements in an ArrayList.  
How would you iterate through the ArrayList and print each employee name?  
  
Look for ways to loop through all elements in an ArrayList.

Q1.Write a Java program to create a list of integers and perform the following operations:  
Add an element to the end of the list.  
Insert an element at a specific position.  
Remove an element from the list.  
Check if a certain element exists in the list.  
  
  
Q2.List Iteration:  
Create a list of strings and iterate through it using a for loop, printing each element.  
  
Q3.List Sorting:  
Write a program to create a list of strings and sort it alphabetically.  
  
Q4.List Searching:  
Create a list of integers and write a method to search for a specific element in the list. Return true if found, otherwise false.  
  
Q5.Liist Filtering:  
Given a list of integers, write a program to filter out even numbers and create a new list with only the odd numbers.  
List Intersection:  
  
Q6.Write a Java program that takes two lists of integers and returns a new list containing only the elements that are common to both lists.  
  
Q7.List Reversal:  
Write a method to reverse a given list of strings in Java.  
  
Q8.List Concatenation:  
Write a program to concatenate two lists of strings into a single list.  
  
Q9.List Removal:  
Given a list of strings, remove all occurrences of a specified element from the list.  
  
Q10.List Element Count:  
Write a method to count the occurrences of a specific element in a given list.

1. Write a program to sort an ArrayList in descending order based on  
   i.  List of Integer  
   ii. List of String  
   iii.List of Employees based on their empID.  
  
 Pass the List of elements as inputs to methods where  
 sorting will be performed.

package pack2;

import java.util.ArrayList;

import java.util.Scanner;

class Employee

{

int empId;

String name;

float sal;

public Employee(int empId, String name, float sal) {

super();

this.empId = empId;

this.name = name;

this.sal = sal;

}

@Override

public String toString() {

return "Employee [empId=" + empId + ", name=" + name + ", sal=" + sal + "]";

}

}

public class MyProgram

{

public static void numShort(ArrayList<Integer> l)

{

for(int i=0;i<l.size()-1;i++)

{

for(int j=i+1;j<l.size();j++)

{

int a=l.get(i);

int b=l.get(j);

if(a>b)

{

l.set(i, b);

l.set(j, a);

}

}//j for ends

}//i for ends

System.out.println(l);

}

public static void nameShort(ArrayList<String> n)

{

for(int i=0;i<n.size()-1;i++)

{

for(int j=i+1;j<n.size();j++)

{

String a=n.get(i);

String b=n.get(j);

if(a.charAt(0)>b.charAt(0))

{

n.set(i,b);

n.set(j, a);

}

}//j for ends

}//i for ends

System.out.println(n);

}

public static void empShort(ArrayList<Employee> e)

{

for(int i=0;i<e.size()-1;i++)

{

for(int j=i+1;j<e.size();j++)

{

Employee em1= e.get(i);

Employee em2=e.get(j);

if(em1.empId>em2.empId)

{

e.set(i, em2);

e.set(j, em1);

}

}//j

}//i

System.out.println(e);

}

public static void main(String[] args) throws InterruptedException

{

ArrayList<Integer> num=new ArrayList<Integer>();

ArrayList<String> name=new ArrayList<String>();

ArrayList<Employee> emp=new ArrayList<Employee>();

System.out.print("Enter the size of numbers you want to add into list:");

Scanner sc=new Scanner(System.in);

int size=sc.nextInt();

for(int i=0;i<size;i++)

{

num.add(sc.nextInt());

}

numShort(num);

Thread.sleep(2000);

///////////////////////////////////////////////////////////////////////

System.out.print("Enter size of an list to add names : ");

size=sc.nextInt();

for(int i=0;i<size;i++)

{

name.add(sc.next());

}

nameShort(name);

Thread.sleep(500);

//////////////////////////////////////////////////////

// Employee emp1=new Employee(sc.nextInt(),sc.next(),sc.nextFloat());

System.out.print("Enter the no of employees detail do you want to eneter :");

size=sc.nextInt();

for(int i=0;i<size;i++)

{

System.out.println("Enter empid, name,sal of "+(i+1)+" employee :");

Employee em=new Employee(sc.nextInt(),sc.next(),sc.nextFloat());

emp.add(em);

}

empShort(emp);;

}

}

   
2. WAP to create a customised ArrayList where the capacity of the  
   ArrayList object is decided by the user.  
   Store the integers and pass all the integers together as a single  
   unit and separate the even & odd numbers from the list.  
  
   Store separated elements in two different List objects.  
   And print them individually using ListIterator cursor object.  
  
3. You are assigned an application to be developed for NareshIT.  
The application should manage student records, including adding  
new students, removing students, and listing all students.  
Each student has a unique student ID,name,course,noOfMonthCompleted,.  
syllabusCompleted. Make sure you have to follow encapsulation.  
  
  
  
Design a class NareshIT with member functions:  
A List object as an instance member representing list of students.  
  
 -> Design a method addStudent  
    This method will take a Student object as an input to add into the student list.  
  
 -> Design a method removeStudentBasedOnid  
    This method will take a int id as an input to remove that Student object  
    from student list.  
  Note: While removing a student from the record, if student data is  
  not found then throw a customized exception  
  "StudentNotAvailableExecption".  
       
 -> Design a method displayDetails  
    ->This method will display all the student details from the student list.  
  
- Design a method 'getPlacementStudents'  
  If the student has completed 80% of the course or student has  
  been regular in the institute for more than and equal to 6 months  
  then only the student is eligible for placement process.  
  Your job is to return all the students with their details who  
  are eligible for placement process.  
  
- Design a method 'getStudentsBasedOnCourse'  
  This method will recieve an input as course .  
- Some students have joined in different courses.  
  Your job is to return group of students who are enrolled based on  
  given course.  
  If the given course is not in the records for student then  
  throw an exception "NoCourseFoundException"  
  
   
[A list of student may contain 'null',hence it has to be handled]  
[A real java developer never writes all the logic in one single  
class.]  
  
Test the above application to show all the method functionality.  
Follow proper naming convention, encapsulation, and exception handling.

Program: 1  
----------------  
Write a Java program to convert Array to List.  
A BLC class called ArrayToList is given to you.  
Add a static method: convertToList(String[] inputArray) : This method  
takes an array of strings as an argument. It should convert the given  
array to a list of Strings and return this list.  
Note that the length of the list must be the same as that of the array.  
Return null for errors a null array or an empty list for an empty  
array.  
An ELC Class called Tester with the main method is given to you. Use  
this class to test your solution

**package pack2;**

**import java.util.ArrayList;**

**import java.util.List;**

**import java.util.Scanner;**

**class ArrayToList**

**{**

**public static List convertTolist(String [] inputArray)**

**{**

**ArrayList al=new ArrayList(inputArray.length);**

**for(int i=0;i<inputArray.length;i++)**

**{**

**al.add(inputArray[i]);**

**}**

**if(inputArray==null)**

**return null;**

**else**

**return al;**

**}**

**}**

**public class Tester**

**{**

**public static void main(String[] args)**

**{**

**System.*out*.print("Enter the size of an array :");**

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

**int size=sc.nextInt();**

**String [] inputArray=new String [size];**

**for(int i=0;i<size;i++)**

**{**

**inputArray[i]=sc.next();**

**}**

**ArrayList al=(ArrayList) ArrayToList.*convertTolist*(inputArray);**

**System.*out*.println(al);**

**System.*out*.println(al.size());**

**}**

**}**

Program - 2  
------------------------  
Write a java program to find duplicates in an arrayList.  
A BLC class Duplicates is given to you. Add the following  
implementations in the class:  
1. A List attribute called numberList. This must be a list of Integers  
3  
2. Getter/Setter for the list  
3. Implement the No Argument constructor.  
 Implement the parameter constructor.  
 Implement the method getDuplicatesList():  
4. This method must scan through the numberList field of the class.  
All the numbers in the list that are duplicated  
 (occurs more than once) must be returned as a new List.  
 For example, if the numberList is initialized to {  
33,45,67,89,33,47,21,45}, the method should return a new list that  
contains { 33,45}.  
 Note that a number may be repeated any number of times in the  
list, but it should be present in the returned list only once.  
 For example, if the numberList is initialized to { 66,66,77,66,66},  
the method should return a new list that contains { 66}.  
5. If the numberList is empty, or null, return null.

**package pack2;**

**import java.util.ArrayList;**

**import java.util.List;**

**import java.util.Scanner;**

**class FindDublicates**

**{**

**List<Integer> numberList=new ArrayList();**

**public List<Integer> getNumberList() {**

**return numberList;**

**}**

**public void setNumberList(List<Integer> numberList) {**

**this.numberList = numberList;**

**}**

**public FindDublicates() {**

**super();**

**}**

**public FindDublicates(List<Integer> numberList) {**

**super();**

**this.numberList = numberList;**

**}**

**public List getDuplicatesList()**

**{**

**List newList=new ArrayList();**

**for(int i=0;i<numberList.size()-1;i++)**

**{**

**int count=0;**

**for(int j=i+1;j<numberList.size();j++)**

**{**

**if(numberList.get(i)==numberList.get(j))**

**{**

**count++;**

**}**

**if(count==1)**

**newList.add(numberList.get(i));**

**}**

**}**

**if(numberList==null)**

**return null;**

**else**

**return newList;**

**}**

**}**

**public class Tester**

**{**

**public static void main(String[] args)**

**{**

**List l=new ArrayList();**

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

**System.*out*.print("Enter the size of an list :");**

**int size=sc.nextInt();**

**for(int i=0;i<size;i++)**

**{**

**l.add(sc.nextInt());**

**}**

**FindDublicates fd=new FindDublicates(l);**

**System.*out*.println(fd.getDuplicatesList());**

**System.*out*.println(new FindDublicates().getDuplicatesList());**

**}**

**}**

Collections mcq-2  
1)Write a program to convert a given array into a collection with the asList() method.  
String[] str = {"Abc","Mno","Prq"};  
  
  
2)How to Make a Collection Read-Only in Java?  
List = [1,2,3,4,5,6]  
1.Read only List object  
2.Read only Set object  
[Read only means modification is not allowed]  
  
  
3)Java Program to Compare Elements in a Collection.  
Input : List = [3, 5, 18, 4, 6]  
Output: Min value of our list : 3  
Max value of our list : 18  
  
Input : List = [4,4,4,4,4]  
Output : All elements are same.  
  
4)Java Program to Remove a Specific Element all occurence From a Collection.  
List = [1,2,2,3,4,2,5,2]  
element : 2  
List = [1,3,4,5]  
  
5)How to remove an element from ArrayList in Java?  
6)How to Replace an Element in Java ArrayList?  
1.Replace one old element to new element  
List = [1,2,3,2,4,5]  
oldEle : 2  
newEle : 20  
List = [1,20,3,2,4,5]  
  
2.Replace all the ocurrence of old ele to new ele.  
List = [1,2,3,2,4,5]  
oldEle : 2  
newEle : 20  
List = [1,20,3,20,4,5]  
  
7)Compare two ArrayList In Java?  
List1 = [1,2,3]  
List2 = [1,2,3]  
Output : true  
  
List1 = [6,2,3]  
List2 = [1,2,3]  
Output : false  
  
8)Find the prime integers from a List of Integers and print it.  
List = [1,2,3,4,5]  
Output : 2 3 5  
  
9)Java Program to Convert ArrayList to LinkedList.  
  
10)Java Program to Convert Array to LinkedList in Java.  
11)How to Sort HashSet in Java.  
12)Convert Array to HashSet in Java.  
13)Store n times "Java" in collection object.  
n -> 5  
List = ["Java", "Java", "Java", "Java"]  
  
14)Find how many times a given number appeared in list and print the count.  
List = [1,2,1,2,3,1,3,1]  
Count of 1 : 4  
  
-> Print all the occurence of each element of the above list.  
1 : 4 times  
2 : 2 times  
3 : 3 times  
  
15)Check the two collection objects contains unique elements.  
List = [1,2,3,4]  
Set = [6,7,8,9]  
Output : They both contains different objects  
16)Rotate array list elements n times.  
List = [1,2,3,4,5,6]  
Rotate : 4  
Output : [4, 5, 6, 1, 2, 3]  
  
17)Swap two index element in a collection object.  
List = [1,2,3,4,5,6]  
index1 = 2  
index2 = 4  
Output : [1,2,5,4,3,6]  
18)Reverse the elements in collection object.  
  
19)Print all the even and odd elements separately from a List of Integers.  
20)Print only common integers from two ArrayList of Integers.  
List1 = [1,2,3,4,5]  
List2 = [1,2,6,7,3]  
  
Output : [1, 2, 3]  
  
21)Print only uncommon integers from two ArrayList of Integers.  
List1 = [1,2,3,4,5]  
List2 = [1,2,6,7,3]  
  
Output : [5,6,7]  
  
22)Check the List is having at least one same element or all different elements.  
List1 = [1,2,3,4,5]  
List2 = [1,2,6,7,3]  
Output : Two list is having atleast one same element  
  
List1 = [1,2,3,4,5]  
List2 = [12,13,14,15]  
Output : Two list is having all different elements.  
  
  
23)Given a list of strings, write a Java program to filter out and print all strings that contain the letter "a".  
24)Write a program to convert all the strings in a list to uppercase and print the result.  
25)Given two lists of integers, write a program to combine them into a single list and then find and print the sum of all elements in the combined list.

Q1)What is the output of the following code?

public class CollectionExample {

public static void main(String[] args) {

TreeSet hash = new TreeSet < > ();

hash.add(“nit”);

hash.add(“NIT”);

System.out.println(hash);

System.out.println(hash.size());

}

}

A. NIT nit 2

B. nit 1

C. [NIT] 1

D. [nit NIT] 2

Question: 2

class Employee{

int id;

String name;

public Employee(int id, String name) {

super();

this.id = id;

this.name = name;

}

@Override

public String toString() {

return “Employee [id=” + id + “, name=” + name + “]”;

}

}

public class CollectionProgram004 {

public static void main(String[] args) {

Vector < Object > v1=new Vector();

v1.add(10);

v1.add(new Employee(1,”abc”));

v1.add(“hello”);

System.out.print(v1.removeElement(10)+" ");

System.out.print(v1.removeElement(new Employee(1,"abc")));

}

}

A. true false

B. false false

C. true true

D. None of the above

Question: 3

public static Collection get() {

Collection sorted = new LinkedList();

sorted.add(“B”); sorted.add(“C”); sorted.add(“A”);

return sorted;

}

public static void main(String[] args) {

for (Object obj: get()) {

System.out.print(obj + “, “);

}

}

What is the result?

Options:-

A. A, B, C,

B. B, C, A,

C. Compilation fails.

D. The code runs with no output.

E. An exception is thrown at runtime.

Question: 4

public static void main(String[] args) {

boolean[] ba = new boolean[4];

Set s = new HashSet();

ba[0] = s.add(“a”);

ba[1] = s.add(new Integer(42));

ba[2] = s.add(42);

ba[3] = s.add(new StringBuffer(“a”));

for(int x=0; x < ba.length; x++)

System.out.print(ba[x] + ” “);

}

}

Options:-

A.true true false true

B.true true true false

C.a 42 42 a

D.a 42 a

Question: 5

public class WrappedString {

private String s;

public WrappedString(String s) {

this.s = s;

}

public static void main(String[] args) {

HashSet < Object > hs = new HashSet < Object > ();

WrappedString ws1 = new WrappedString(“A”);

WrappedString ws2 = new WrappedString(“A”);

String s1 = new String(“A”);

String s2 = new String(“A”);

hs.add(ws1); hs.add(ws2); hs.add(s1); hs.add(s2);

System.out.println(hs.size()); } }

What is the result?

Options:-

A. 1

B. 2

C. 3

D. 4

Question: 6

Which two statements are true about the hashCode method? (Choose two.)

A.) The hashCode method for a given class can be used to test for object equality and object inequality for that class.

B) The hashCode method is used by the java.util.SortedSet collection class to order the elements within that set.

C) The hashCode method for a given class can be used to test for object inequality, but NOT object equality, for that class.

D) The only important characteristic of the values returned by a hashCode method is that the distribution of values must follow a Gaussian distribution.

E. The hashCode method is used by the java.util.HashSet collection class to group the elements within that set into hash buckets for swift retrieval.

Options:-

i)A ,E

ii)A ,C

iii)B ,D

iv)D ,E

V)C,E

Question: 7

import java.util.\*;

4. class Turtle {

5. int size;

6. public Turtle(int s) { size = s; }

7. public boolean equals(Object o) { return (this.size == ((Turtle)o).size); }

8. // insert code here

9. }

10. public class TurtleTest {

11. public static void main(String[] args) {

12. LinkedHashSet < Turtle > t = new LinkedHashSet < Turtle > ();

13. t.add(new Turtle(1)); t.add(new Turtle(2)); t.add(new Turtle(1));

14. System.out.println(t.size());

15. }

16. }

And these two fragments:

I. public int hashCode() { return size/5; }

II. // no hashCode method declared

If fragment I or II is inserted independently at line 8, which are true? (Choose all that apply.)

A. If fragment I is inserted, the output is 2

B. If fragment I is inserted, the output is 3

C. If fragment II is inserted, the output is 2

D. If fragment II is inserted, the output is 3

E. If fragment I is inserted, compilation fails

Options:-

i) A&B

ii) C&D

iii)B&E

iv)A&E

V)A&D

Question: 8

What is the output?

public class CollectionExample {

public static void main(String[] args) {

TreeSet set = new TreeSet < > ();

set.add(“NIT”);

set.add(“nit”);

set.add(5.0);

set.add(5.00);

set.add(1);

set.add(1);

Iterator it = set.iterator();

while(it.hasNext())

{

System.out.print(it.next()+” “);

}

}

}

Options:-

A. 1 5.0 5.00 NIT nit

B. 1 1 5.0 5.00 NIT nit

C. 1 5.0 NIT nit

D. 1 5.0 NIT

Compiler time Error

Question: 9

public class Drink implements Comparable {

public String name;

public int compareTo(Object o) {

return 0;

}

}

and:

Drink one = new Drink();

Drink two = new Drink();

one.name= “Coffee”;

two.name= “Tea”;

TreeSet set = new TreeSet();

set.add(one);

set.add(two);

A programmer iterates over the TreeSet and prints the name of each Drink object.

What is the result?

Options:-

A. Tea

B. Coffee

C. Coffee Tea

D. Compilation fails.

E. The code runs with no output.

Question: 10

public class CollectionProgram003 {

public static void main(String[] args) {

LinkedList list = new LinkedList();

list.add(new Integer(2));

list.add(new Integer(8));

list.add(new Integer(5));

list.add(new Integer(1));

Iterator i = list.iterator();

Collections.reverse(list);

Collections.sort(list);

while (i.hasNext())

System.out.print(i.next() + " ");

}

}

Options:-

A. 8 5 2 1

B. 1 2 5 8

C. IndexOutOfBoundException

D. 2 8 5 1

Collections mcq-3

Question: 1

What is the output of the following code?

public class Testing {

public static void main(String[] args) {

Map < String,String > m=new HashMap < String, String > ();

if(m.put("Aniket", "1234") != null)

{

System.out.println(m.put("Aniket", "1234") );

}

else

{

System.out.println("Not Insetred");

}

}

}

Options:-

A)Data inserted

B)Not Insetred

c)Compiler time Error Because put Return type is boolean

d)Runtime Error because same key we are inserting again

Question: 2

What is the output of the following code?

public class Testing {

public static void main(String[] args) {

ArrayList < String > al=new ArrayList < String > ();

if(al.add("Anu\_786")!=null)

{

System.out.println("Inserted");

}

else

{

System.out.println("Not Insetred");

}

}

}

Options:-

A) Inserted

b)Not Insetred

c)Compiler time Error

d) Runtime Error

Question: 3

What is the output of the following code?

public class Testing {

public static void main(String[] args) {

Map < Integer,Integer > m=new HashMap < Integer, Integer > ();

m.put(1, 121);

m.put(1, 123);

m.put(111, 111);

m.put(1, 134);

System.out.println(m);

}

}

Options:-

A) {1=134, 111=111}

b) {1=121, 111=111}

c) {1=121, 1=123, 111=111, 1=134}

d) Runtime Error

Question: 4

What is the output of the following code?

public class Testing {

public static void main(String[] args) {

Map < String,Character > m=new HashMap < String, Character > ();

m.put("123", 'A');

m.put("124", 'B');

m.put("125", 'C');

m.put("121", 'D');

Set s1=m.keySet();

System.out.println(s1);

}

}

Options:-

A) [121=D, 123=A, 124=B, 125=C]

b) [121, 123, 124, 125]

c) {121=D, 123=A, 124=B, 125=C}

d) Compiler time Error Because KeySet parameter Missing

e) Runtime Error

Question: 5

What is the use of entrySet() method ?

Options:-

A) To Convert Map to Set

b) returns a set view of all the entries from the hashmap.

c) To retrive the key by using entryset()

d) None of the above

Question: 6

What is the output of the following code?

public class Testing {

public static void main(String[] args) {

Map < String,String > m=new HashMap < String, String > ();

m.put("Aniket", "1224");

m.put("Amarjeet", "1225");

m.put("Biren", "1226");

m.put("Shankar", "1227");

m.put("Aniket", "1228");

if(m.containsKey("Aniket"))

{

System.out.println(m.get("Aniket"));

}

else

{

System.out.println("Data Not Found");

}

}

}

Options:-

A) Compiler Time Error

b) Executed Successfully..! Data Not Found

c) 1228

d) 1224

e) Data Not Found

Question: 7

What is the output of the following code?

public class Testing {

public static void main(String[] args) {

Map < String,String > m=new HashMap < String, String > ();

m.put("Aniket", "1224");

m.put("Amarjeet", "1225");

m.put("Biren", "1226");

m.put("Shankar", "1227");

if(m.containsKey("1224"))

{

System.out.println("Data Not Found");

}

else

{

System.out.println("Data Found");

}

}

}

Options:-

A) Compiler Time Error

b) Runtime Error

c) Data Not Found

d)Data Found

Question: 8

What is the output of the following code?

public class Testing {

public static void main(String[] args) {

Map < String,String > m=new HashMap < String, String > ();

m.put("Aniket", "1224");

m.put("Amarjeet", "1225");

m.put("Biren", "1226");

m.put("Aniket", "1221");

m.put("Shankar", "1227");

m.put("Aniket", "1228");

if(m.size()>m.size()-1)

{

System.out.println(m);

}

else

{

System.out.println("finally you got the answer");

}

}

}

Options:-

A) {Aniket=1228, Amarjeet=1225, Biren=1226, Shankar=1227}

b) {Aniket=1224, Amarjeet=1225, Biren=1226, Shankar=1227}

c) {Aniket=1221, Amarjeet=1225, Biren=1226, Shankar=1227}

d) finally you got the answer

e) {Aniket=1224, Amarjeet=1225, Biren=1226, Aniket=1221, Shankar=1227, Aniket=1228}

Question: 9

Given:

3. import java.util.\*;

4. public class Mapit {

5. public static void main(String[] args) {

6. Set < Integer > set = new HashSet < Integer > ();

7. Integer i1 = 45;

8. Integer i2 = 46;

9. set.add(i1);

10. set.add(i1);

11. set.add(i2); System.out.print(set.size() + ” “);

12. set.remove(i1); System.out.print(set.size() + ” “);

13. i2 = 47;

14. set.remove(i2); System.out.print(set.size() + ” “);

15. }

16. }

What is the result?

Options:-

A. 2 1 0

B. 2 1 1

C. 3 2 1

D. 3 2 2

E. An exception is thrown at runtime.

Question: 10

public class Testing {

public static void main(String[] args) {

Map < String, String > map = new HashMap < > ();

map.put(“123”, “Anu”);

map.put(“124”, “Hemanth”);

map.put(“125”, “Sagar”);

map.put(“124”, “pratik”);

map.put(“126”, “rahul”);

if(map.entrySet().isEmpty()!=0)

{

System.out.println(true);

}

else

{

System.out.println(false);

}

}

}

Options:-

A) True

B) false

C) Compilation error

D) Runtime error

Batch 27 task -9

Program 01 :

------------

Create a record called Product which is accepting two components pid and pname.

Write one compact constructor to throw IllegalArgumentException if pid is zero or Negative.

Create one TreeSet class which will hold the Custom product object.

With the help of TreeSet constructor using Comparator interface, write the logic to sort the Product data on the following basis

1) Sort based on the pid in ascending order.

2) Sort based on the pid in descending order.

3) Sort based on the name in Alphabetical order.

4) Sort based on the name in Reverse order.

----------------------------------------------------------------------

Program 02 :

------------

Create a record called Book which take two components name and author.

Implement this Book record with Comparable interface to sort the books based on the name using compareTo() method.

Create an ELC class BookSorted which will sort the book based on the name.

Create one TreeSet object which will hold Book as a custom object.

Insert 5 book records using custom object.

Display all the books sorted in alphabetial order.

Batch 29 task on collection map mcq

Q1) Find the output

public class HashMapInternals

{

public static void main(String[] args)

{

Map<String,String> map = new HashMap<>();

map.put("Raj","Ampt");

map.put(new String("Raj"),"Hyd");

System.out.println(map);

}

}

Q2) Find the output

public class BufferTest

{

public static void main(String [] args)

{

Integer i1 = 128;

Integer i2 = 128;

System.out.println(i1==i2);

System.out.println(i1.equals(i2));

}

}

Q3) Find the output

public class BufferTest

{

public static void main(String [] args)

{

Map<Integer,String> map = new HashMap<>();

map.put(128,"Ampt");

map.put(new Integer(128),"Hyd");

System.out.println(map.size());

}

}

Q4) Find the output

public class CollectionExample

{

public static void main(String[] args)

{

HashMap hash = new HashMap<>();

hash.put("nit", 500);

hash.put("info", 644);

hash.put("tech", "google");

System.out.println(hash.get("tech"));

System.out.println(hash.get("google"));

System.out.println(hash.getOrDefault("nit", "Technology"));

System.out.println(hash.get("info"));

}

}

Q5) Find the output

public class CollectionExample

{

public static void main(String[] args)

{

HashMap hash = new HashMap<>();

hash.put("nit", 500);

hash.put("info", 644);

hash.put("tech", "google");

System.out.println(hash.get("tech"));

System.out.println(hash.get("google"));

System.out.println(hash.getOrDefault("nit", "Technology"));

System.out.println(hash.get("info"));

}

}

Q6) Find the output

public class HashMapExample

{

public static void main(String[] args)

{

Map<String, Integer> map = new HashMap<>();

map.put("apple", 1);

map.put("banana", 2);

map.put("orange", 3);

map.replaceAll((k, v) -> v \* 2);

System.out.println(map);

}

}

Q7) Find the output

public class Participate

{

public static void main(String args[]) {

HashMap participant = new HashMap();

participant.put(1 + 1, "Dhoni");

participant.put(1 + 1, "Kohli");

participant.put(2 + 1, "Rohit");

Set set = participant.entrySet();

Iterator itr = set.iterator();

while (itr.hasNext()) {

Map.Entry m = (Entry) itr.next();

System.out.print(m.getKey() + " " + m.getValue()+" ");

}

}

Q8) Find the output

import java.util.HashMap;

import java.util.Map;

public class Test

{

public static void main(String[] args)

{

Map<ToDos, String> m = new HashMap<ToDos, String>();

ToDos t1 = new ToDos("Monday");

ToDos t2 = new ToDos("Monday");

ToDos t3 = new ToDos("Tuesday");

m.put(t1, "doLaundry");

m.put(t2, "payBills");

m.put(t3, "cleanAttic");

System.out.println(m.size());

}

}

class ToDos

{

String day;

ToDos(String d)

{

day = d;

}

public boolean equals(Object o)

{

return ((ToDos) o).day == this.day;

}

//Comment hashcode method and see and analyse the result

public int hashCode() { return 9; }

}

Q9) Find the output

public class Test

{

public static void main(String[] args)

{

NavigableMap<Integer, String> map = new TreeMap<Integer, String>();

map.firstKey();

System.out.println(map.size());

}

}

10) Find the output

public class Test

{

public static void main(String[] args)

{

Map<String, Integer> map = new HashMap<String, Integer>();

map.put("T", 1);

map.put("M", 2);

map.keySet().add("A",3);

System.out.println(map.size());

}

}

Batch 26 task-9

Program 01

-----------

You have to create a simple Student Grade Book application in Java. The program should allow users to maintain a list of students along with their grades, add new students and grades, remove existing students, and exit the application.

You are provided with a partially implemented Java program for the Student Grade Book. However, it has some issues, and your task is to complete the implementation and ensure that it works correctly.

Requirements:

The program should display the following menu to the user:

Student Grade Book:

(List of students and their grades displayed here)

Options:

1. Add Student Grade

2. Remove Student Grade

3. Exit

Enter your choice:

If the grade book is empty, it should indicate that there are no students and grades to display.

Implement the following functionality for each menu option:

Add Student Grade (Option 1): Allow the user to enter the name of a new student and their grade. Add this information to the grade book as a key-value pair (student name as the key and grade as the value). Print a confirmation message indicating that the grade has been added for the student.

Remove Student Grade (Option 2): If the grade book is empty, display a message indicating that it's empty. Otherwise, allow the user to enter the name of the student whose grade they want to remove. If the student exists in the grade book, remove the student and their grade from the grade book and print a confirmation message. If the student is not found, print an error message.

Exit (Option 3): Display a farewell message and exit the program.

For any other invalid input, display a message indicating that the choice is invalid and ask the user to select a valid option.

The program should run in a continuous loop until the user selects the "Exit" option.

Your Task:

Complete the implementation of the provided Java code.

Ensure that the program handles user input correctly and provides the expected functionality as described in the requirements.

Test the program thoroughly to ensure it works as expected.

Submit the complete Java code for the Student Grade Book application.

Note: You can use the provided Java code as a starting point and make any necessary modifications to meet the requirements of the problem statement.

**package com.nit.collections;**

**//Student Grade Book application**

**import java.util.HashMap;**

**import java.util.Map;**

**import java.util.Scanner;**

**public class StudentGardeBook**

**{**

**public Map<String,String> gradeBook;**

**public StudentGardeBook()**

**{**

**gradeBook=new HashMap();**

**}**

**public void displayMenu()**

**{**

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

**System.*out*.println("Student Grade Book:");**

**if(gradeBook.isEmpty())**

**{**

**System.*out*.println("No students and grades to display.");**

**}**

**else**

**{**

**for(Map.Entry<String, String> entry:gradeBook.entrySet())**

**{**

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

**}**

**}**

**System.*out*.println("\nOptions:\r\n"**

**+ "1. Add Student Grade\r\n"**

**+ "2. Remove Student Grade\r\n"**

**+ "3. Exit\r\n"**

**+ "");**

**}//displayMenu()**

**public void addStudentGrade(Scanner sc)**

**{**

**sc.nextLine();**

**System.*out*.print("Enter Student Name : ");**

**String name=sc.nextLine();**

**System.*out*.print("Enter Student grade : ");**

**String grade=sc.nextLine();**

**gradeBook.put(name, grade);**

**System.*out*.println("Grade added for student : "+name);**

**}//addStudent**

**public void removeStudentGrade(Scanner sc)**

**{**

**if(gradeBook.isEmpty())**

**{**

**System.*out*.println("The grade book is empty :");**

**return;**

**}**

**System.*out*.println("Enter Student name to remove :");**

**sc.nextLine();**

**String name=sc.nextLine();**

**if(gradeBook.containsKey(name))**

**{**

**gradeBook.remove(name);**

**System.*out*.println("Grade removed for Student :"+name);**

**}**

**else**

**{**

**System.*out*.println("Student not found. ");**

**}**

**}//removeStudent**

**public static void main(String[] args)**

**{**

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

**StudentGardeBook gradeBook=new StudentGardeBook();**

**while(true)**

**{**

**gradeBook.displayMenu();**

**System.*out*.println("Enter your choice: ");**

**int choice=sc.nextInt();**

**switch(choice)**

**{**

**case 1:**

**gradeBook.addStudentGrade(sc);**

**break;**

**case 2:**

**gradeBook.removeStudentGrade(sc);**

**break;**

**case 3:**

**System.*out*.println("Goodbye!");**

**sc.close();**

**System.*exit*(0);**

**break;**

**default :**

**System.*out*.println("Invalid choice .please enter a valid option . ");**

**break;**

**}**

**}**

**}**

**}**

Program 02

----------

You have to create an Address Book application in Java that utilizes a Contact class to represent contacts. The program should allow users to maintain an address book, add new contacts, remove existing contacts, and exit the application.

You are provided with a Java program for the Address Book application. The program includes a Contact class that represents contacts, and it has some issues. Your task is to complete the implementation and ensure that it works correctly.

Requirements:

The program should display the following menu to the user:

Address Book:

(List of contacts displayed here)

Options:

1. Add Contact

2. Remove Contact

3. Exit

Enter your choice:

If the address book is empty, it should indicate that the address book is empty.

Implement the following functionality for each menu option:

Add Contact (Option 1): Allow the user to enter the contact's name and their phone number. Create a new Contact object with the provided information and add it to the address book using the name as the key and the Contact object as the value. Print a confirmation message indicating that the contact has been added.

Remove Contact (Option 2): If the address book is empty, display a message indicating that it's empty. Otherwise, allow the user to enter the name of the contact to remove. If the contact name is found in the address book, remove the contact and print a confirmation message. If the contact name is not found, print an error message.

Exit (Option 3): Display a farewell message and exit the program.

For any other invalid input, display a message indicating that the choice is invalid and ask the user to select a valid option.

The program should run in a continuous loop until the user selects the "Exit" option.

Your Task:

Complete the implementation of the provided Java code.

Ensure that the program handles user input correctly and provides the expected functionality as described in the requirements.

Test the program thoroughly to ensure it works as expected.

Submit the complete Java code for the Address Book application.

Note: You can use the provided Java code as a starting point and make any necessary modifications to meet the requirements of the problem statement.