**Java Assignment 01**

1. **Collection Coding questions:**

**Que 01 – Write a Java program to create a new array list, add some colors (string) and print out the collection.**

**Ans🡪**

import java.lang.\*;

import java.util.\*;

class ArrayListColors1{

public static void main(String args[]){

ArrayList<String> l1 = new ArrayList<String>();

l1.add("Red");

l1.add("Blue");

l1.add("Black");

l1.add("Brown");

l1.add("Yellow");

l1.add("Orange");

for(String x : l1){

System.out.println(x);

}

}

}

**==============================================================**

**Que 02 – Write a Java program to insert an element into the array list at the first position.**

**Ans🡪**

import java.lang.\*;

import java.util.\*;

class ArrayListAddElement2{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

ArrayList<String> l1 = new ArrayList<String>();

l1.add("Red");

l1.add("Blue");

l1.add("Black");

l1.add("Brown");

l1.add("Yellow");

l1.add("Orange");

System.out.println("Original array list is : ");

for(String x : l1){

System.out.print(x+" ");

}

System.out.println();

System.out.println("Enter the element you want to add : ");

String s = sc.nextLine();

l1.add(0, s);

System.out.println();

System.out.println("Array list after adding element is : ");

for(String x : l1){

System.out.print(x+" ");

}

}

}

**==============================================================**

**Que 03 – Write a Java program to retrieve an element (at a specified index) from a given array list.**

**Ans🡪**

import java.lang.\*;

import java.util.\*;

class ArrayListRetriveElement3{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

ArrayList<String> l1 = new ArrayList<String>();

l1.add("Red");

l1.add("Blue");

l1.add("Black");

l1.add("Brown");

l1.add("Yellow");

l1.add("Orange");

System.out.println("Original array list is : ");

for(String x : l1){

System.out.print(x+" ");

}

System.out.println();

System.out.println("Enter the index of the element you want to retrive : ");

int s = sc.nextInt();

System.out.println("Element you want is : "+l1.get(s));

}

}

**==============================================================**

**Que 04 – Write a Java program to sort a given array list.**

**Ans🡪**

import java.lang.\*;

import java.util.\*;

class ArrayListSorting4{

public static void main(String args[]){

ArrayList<String> l1 = new ArrayList<String>();

l1.add("Red");

l1.add("Blue");

l1.add("Black");

l1.add("Brown");

l1.add("Yellow");

l1.add("Orange");

System.out.println("Original array list is : ");

for(String x : l1){

System.out.print(x+" ");

}

Collections.sort(l1);

System.out.println();

System.out.println("Array list after sorting : ");

for(String x : l1){

System.out.print(x+" ");

}

}

}

**==============================================================**

**Que 05 – Write a Java program to reverse elements in a array list.**

**Ans🡪**

import java.lang.\*;

import java.util.\*;

class ArrayListReverse5{

public static void main(String args[]){

ArrayList<String> l1 = new ArrayList<String>();

l1.add("Red");

l1.add("Blue");

l1.add("Black");

l1.add("Brown");

l1.add("Yellow");

l1.add("Orange");

System.out.println("Original array list is : ");

for(String x : l1){

System.out.print(x+" ");

}

Collections.reverse(l1);

System.out.println();

System.out.println("Array list after reversing : ");

for(String x : l1){

System.out.print(x+" ");

}

}

}

**==============================================================**

**Que 06 – Write a Java program of swap two elements in an array list.**

**Ans🡪**

import java.lang.\*;

import java.util.\*;

class ArrayListSwappingTwoElements6{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

ArrayList<String> l1 = new ArrayList<String>();

l1.add("Red");

l1.add("Blue");

l1.add("Black");

l1.add("Brown");

l1.add("Yellow");

l1.add("Orange");

System.out.println("Original array list is : ");

for(String x : l1){

System.out.print(x+" ");

}

System.out.println();

System.out.println("Enter the elements you want to swap : ");

String s1 = sc.nextLine();

String s2 = sc.nextLine();

int n1 = l1.indexOf(s1);

int n2 = l1.indexOf(s2);

l1.remove(s1);

l1.remove(s2);

l1.add(n1, s2);

l1.add(n2, s1);

System.out.println("Array list after Swapping : ");

for(String x : l1){

System.out.print(x+" ");

}

}

}

**==============================================================**

**Que 07 – Write a Java program to print all the elements of a ArrayList using the position of the elements.**

**Ans🡪**

import java.util.\*;

class Q7

{

public static void main(String args[])

{

ArrayList<String> list1=new ArrayList<String>();

list1.add("Rohit"); //0

list1.add("Mukesh"); //1

list1.add("Radha"); //2

list1.add("Geeta"); //3

list1.add("Shubham"); //4

System.out.println(list1.get(2)); //print by position

}

}

/\*

OUTPUT :

Radha

\*/

**==============================================================**

**Que 08 – Write a Java program to append the specified element to the end of a linked list.**

**Ans🡪**

class Q8List

{

static Node head;

static class Node{

int data;

Node next;

Node(int d)

{

data = d;

next = null;

}

}

static void insert(int new\_data)

{

Node new\_node = new Node(new\_data);

new\_node.next = head;

head = new\_node;

}

static void append(int new\_data)

{

Node new\_node = new Node(new\_data);

if(head == null)// List is empty

{

head = new Node(new\_data);//new node as 1st node in the list

return;

}

new\_node.next = null;

Node n =head;//naya reference banaya

while(n.next != null)//Loop:jab tak end nahi aata hai...tab tak loop chalega

{

n=n.next;

}

n.next = new\_node;//last link ko connect karenge new node ke sath

return;

}

void display()

{

Node n = head;

while(n != null)

{

System.out.print(n.data+"--->");

n=n.next;

}

}

public static void main(String args[])

{

Q8List l1 = new Q8List();

l1.display();

System.out.println("Empty List !!!");

append(2);

insert(5);

insert(15);

insert(25);

insert(35);

append(22);

append(52);

l1.display();

System.out.println();

l1.display();

System.out.println();

l1.display();

}

}

**==============================================================**

**Que 09 – Write a Java program to insert the specified element at the specified position in the linked list.**

**Ans🡪**

import java.lang.\*;

import java.util.\*;

class LinkedListElementAtSpecifiedPosition9{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

LinkedList<String> l1 = new LinkedList<String>();

l1.add("Red");

l1.add("Blue");

l1.add("Black");

l1.add("Brown");

l1.add("Yellow");

l1.add("Orange");

System.out.println("Original linked list is : ");

for(String x : l1){

System.out.print(x+" ");

}

System.out.println();

System.out.println("Enter the element you want to add : ");

String s = sc.nextLine();

System.out.println("Enter the position where you want to add element : ");

int n = sc.nextInt();

l1.add(n, s);

System.out.println("Linked list after appending element : ");

for(String x : l1){

System.out.print(x+" ");

}

}

}

**==============================================================**

**Que 10 – Write a Java program to insert elements into the linked list at the first and last position.**

**Ans🡪**

import java.lang.\*;

import java.util.\*;

class LinkedListElementAtEndAndFirst10{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

LinkedList<String> l1 = new LinkedList<String>();

l1.add("Red");

l1.add("Blue");

l1.add("Black");

l1.add("Brown");

l1.add("Yellow");

l1.add("Orange");

System.out.println("Original linked list is : ");

for(String x : l1){

System.out.print(x+" ");

}

System.out.println();

System.out.println("Enter the element you want to add at the first : ");

String s1 = sc.nextLine();

System.out.println("Enter the element you want to add at the end : ");

String s2 = sc.nextLine();

l1.add(0, s1);

l1.add(l1.size(), s2);

System.out.println("Linked list after appending elements : ");

for(String x : l1){

System.out.print(x+" ");

}

}

}

**==============================================================**

**Que 11 – Write a Java program to display the elements and their positions in a linked list.**

**Ans🡪**

class Que11\_LinkedListElement\_Position

{

static Node head;

static class Node

{

int data;

Node next;

Node ()

{

}

Node(int d)

{

data =d;

next =null;

}

}

static void display(Node head)

{

Node temp = head;

int pos=0;

if(head==null)

System.out.println("List isempty");

while(temp.next!=null)

{

System.out.println("Pos: "+pos+" Data: "+temp.data+"--->");

temp=temp.next;

pos++;

}

System.out.println();

}

public static void main(String[] args)

{

Que11\_LinkedListElement\_Position list = new Que11\_LinkedListElement\_Position();

Node n1= new Node(2);

Node n2= new Node(4);

Node n3= new Node(6);

Node n4= new Node(8);

Node n5= new Node(10);

Node n6= new Node(12);

Node n7= new Node(14);

head=n1;

n1.next=n2;

n2.next=n3;

n3.next=n4;

n4.next=n5;

n5.next=n6;

n6.next=n7;

System.out.println("Current Linked List :- ");

display(list.head);

// System.out.println("Current Linked List :- ");

}

}

/\***OUTPUT - LinkedList Element and its position**

F:\CDAC\CDAC\_Lab\Diwali Homework\Collection coding questions>java Que11\_LinkedListElement\_Position

Current Linked List :-

Pos: 0 Data: 2--->

Pos: 1 Data: 4--->

Pos: 2 Data: 6--->

Pos: 3 Data: 8--->

Pos: 4 Data: 10--->

Pos: 5 Data: 12--->

\*/

**==============================================================**

**Que 12 – Write a Java program to check if a particular element exists in a linked list**

**Ans🡪**

import java.util.Scanner;

import java.io.\*;

class Que12\_SLL\_ElementCheck

{

static Node head;

static class Node

{

int data;

Node next;

Node(int d)

{

data = d;

next = null;

}

}

static void display(Node head)

{

Node temp=head;

if(head==null)

System.out.println("List is empty");

while(temp.next!=null)

{

System.out.print(temp.data+"--->");

temp=temp.next;

}

System.out.println();

}

static int checkelement(Node head, int key)

{

Node temp=head;

int pos=0;

if(head==null)

System.out.println("List is empty");

while(temp.next!=null)

{

if(temp.data==key)

return pos;

temp=temp.next;

pos++;

}

return -1;

}

public static void main(String[] args)

{

Que12\_SLL\_ElementCheck list= new Que12\_SLL\_ElementCheck();

Node n1= new Node(2);

Node n2= new Node(4);

Node n3= new Node(6);

Node n4= new Node(8);

Node n5= new Node(10);

Node n6= new Node(12);

Node n7= new Node(14);

head=n1;

n1.next=n2;

n2.next=n3;

n3.next=n4;

n4.next=n5;

n5.next=n6;

n6.next=n7;

System.out.println("Current Linked List :- ");

display(list.head);

System.out.println("Enter the element you want to search in the list:- ");

Scanner sc = new Scanner(System.in);

int key=sc.nextInt();

int pos = checkelement(list.head,key);

if(pos==-1)

System.out.println("Given element "+key+" is not present in the linked List.");

else

System.out.println("Given element "+key+" is present in List at position index-"+pos);

System.out.println();

}

}

/\***OUTPUT - SEARCH ELEMENT IN THE LINKED LIST**

F:\CDAC\CDAC\_Lab\Diwali Homework\Collection coding questions>java Que12\_SLL\_ElementCheck

Current Linked List :-

2--->4--->6--->8--->10--->12--->

Enter the element you want to search in the list:-

6

Given element 6 is present in List at position index-2

F:\CDAC\CDAC\_Lab\Diwali Homework\Collection coding questions>java Que12\_SLL\_ElementCheck

Current Linked List :-

2--->4--->6--->8--->10--->12--->

Enter the element you want to search in the list:-

2

Given element 2 is present in List at position index-0

F:\CDAC\CDAC\_Lab\Diwali Homework\Collection coding questions>java Que12\_SLL\_ElementCheck

Current Linked List :-

2--->4--->6--->8--->10--->12--->

Enter the element you want to search in the list:-

3

Given element 3 is not present in the linked List.

F:\CDAC\CDAC\_Lab\Diwali Homework\Collection coding questions>

\*/

**==============================================================**

**Que 13 – Write a Java program to compare two linked lists.**

**Ans🡪**

import java.lang.\*;

import java.util.\*;

class LinkedListComparison13{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

LinkedList<String> l1 = new LinkedList<String>();

l1.add("Red");

l1.add("Blue");

l1.add("Black");

l1.add("Brown");

l1.add("Yellow");

l1.add("Orange");

System.out.println("linked list 1 is : ");

for(String x : l1){

System.out.print(x+" ");

}

System.out.println();

LinkedList<String> l2 = new LinkedList<String>();

l2.add("Red");

l2.add("Blue");

l2.add("Black");

l2.add("Brown");

l2.add("Yellow");

l2.add("Orange");

System.out.println("linked list 2 is : ");

for(String x : l2){

System.out.print(x+" ");

}

System.out.println();

if(l1.containsAll(l2)){

System.out.println("Both the linked lists are equal");

}

else{

System.out.println("Linked lists are not equal");

}

}

}

**==============================================================**

**Que 14 – Write a Java program to replace an element in a linked list.**

**Ans🡪**

import java.lang.\*;

import java.util.\*;

class LinkedListReplaceElement14{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

LinkedList<String> l1 = new LinkedList<String>();

l1.add("Red");

l1.add("Blue");

l1.add("Black");

l1.add("Brown");

l1.add("Yellow");

l1.add("Orange");

System.out.println("Original linked list is : ");

for(String x : l1){

System.out.print(x+" ");

}

System.out.println();

System.out.println("Enter the element you want to add : ");

String s = sc.nextLine();

System.out.println("Enter the position where you want to add element : ");

int n = sc.nextInt();

l1.remove(n);

l1.add(n, s);

System.out.println("Changed linked list is : ");

for(String x : l1){

System.out.print(x+" ");

}

}

}

**==============================================================**

**Que 15 – Write a Java program to iterate through all elements in a hash list.**

**Ans🡪**

import java.lang.\*;

import java.util.\*;

class HashSetIteration15{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

HashSet<String> l1 = new HashSet<String>();

l1.add("Red");

l1.add("Blue");

l1.add("Black");

l1.add("Brown");

l1.add("Yellow");

l1.add("Orange");

System.out.println("Hash Set is : ");

for(String x : l1){

System.out.print(x+" ");

}

}

}

**==============================================================**

**Que 16 –**  **Write a Java program to empty an hash set.**

**Ans🡪**

import java.lang.\*;

import java.util.\*;

class HashSetEmpty16{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

HashSet<String> l1 = new HashSet<String>();

l1.add("Red");

l1.add("Blue");

l1.add("Black");

l1.add("Brown");

l1.add("Yellow");

l1.add("Orange");

System.out.println("Original hash Set is : ");

for(String x : l1){

System.out.print(x+" ");

}

System.out.println();

l1.clear();

System.out.println("Empty hash Set is : ");

for(String x : l1){

System.out.print(x+" ");

}

}

}

**==============================================================**

**Que 17 –Write a Java program to convert a hash set to an array.**

**Ans🡪**

import java.util.\*;

public class HashsetToarray

{

public static void main(String[] args)

{

// Create a empty hash set

HashSet<String> h\_set = new HashSet<String>();

// use add() method to add values in the hash set

h\_set.add("ravi");

h\_set.add("kalankar");

h\_set.add("bhutal");

h\_set.add("bhosle");

h\_set.add("kamble");

h\_set.add("bondre");

System.out.println("Original Hash Set: " + h\_set);

// Creating an Array

String[] new\_array = new String[h\_set.size()];

h\_set.toArray(new\_array);

// Displaying Array elements

System.out.println("Array elements: ");

for(String element : new\_array)

{

System.out.println(element);

}

}

}

**==============================================================**

**Que 18 – Write a Java program to compare two sets and retain elements which are same on both**

**Ans🡪**

CODE:

import java.util.\*;

public class CompareSetsAndRetain

{

public static void main(String[] args)

{

// Create a empty hash set

HashSet<String> h\_set1 = new HashSet<String>();

// use add() method to add values in the hash set

h\_set1.add("Red");

h\_set1.add("Green");

h\_set1.add("Black");

h\_set1.add("White");

System.out.println("Frist HashSet content: "+h\_set1);

HashSet<String>h\_set2 = new HashSet<String>();

h\_set2.add("Red");

h\_set2.add("Pink");

h\_set2.add("Black");

h\_set2.add("Orange");

System.out.println("Second HashSet content: "+h\_set2);

h\_set1.retainAll(h\_set2);

System.out.println("HashSet content:");

System.out.println(h\_set1);

}

}

**==============================================================**

**Que 19 – Write a Java program to create a new tree set, add some colors (string) and print out the tree set.**

**Ans🡪**

import java.lang.\*;

import java.util.\*;

class TreeSetIteration19{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

TreeSet<String> l1 = new TreeSet<String>();

l1.add("Red");

l1.add("Blue");

l1.add("Black");

l1.add("Brown");

l1.add("Yellow");

l1.add("Orange");

System.out.println("Tree Set is : ");

for(String x : l1){

System.out.print(x+" ");

}

}

}

**==============================================================**

**Que 20 – Write a Java program to find the numbers less than 7 in a tree set.**

**Ans🡪**

import java.lang.\*;

import java.util.\*;

class TreeSetNoLessThan720{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

TreeSet<Integer> s1 = new TreeSet<Integer>();

s1.add(10);

s1.add(4);

s1.add(2);

s1.add(35);

s1.add(7);

s1.add(1);

s1.add(45);

s1.add(5);

System.out.println("Elements in the TreeSet are : ");

for(Integer x : s1){

System.out.println(x);

}

System.out.println("Elements in the TreeSet less than 7 are : ");

for(Integer s : s1){

if(s < 7){

System.out.println(s);

}

}

}

}

**==============================================================**

**Que 21 – Write a Java program to remove all the elements from a priority queue.**

**Ans🡪**

// Elements are of Integer type

import java.util.\*;

public class Que21\_QueueRemoveElement

{

// Main driver method

public static void main(String args[])

{

// Creating an empty PriorityQueue by

// creating an object of integer type

PriorityQueue<Integer> queue = new PriorityQueue<Integer>();

// Adding custom input elements

// using add() method

queue.add(10);

queue.add(15);

queue.add(30);

queue.add(20);

queue.add(5);

// Displaying the PriorityQueue

System.out.println("Initial PriorityQueue: "+ queue);

// Removing elements from the PriorityQueue

// using remove() method - one element at a time

queue.remove(30);

queue.remove(5);

// Displaying the PriorityQueue elements after removal

System.out.println("PriorityQueue after removing elements: " + queue);

queue.clear(); // method to remove all elements in queue

// Displaying the PriorityQueue elements after all elements removal

System.out.println("PriorityQueue after removing all elements(clear() method use): " + queue);

}

}

/\***OUTPUT - REMOVING ELEMENTS FROM PRIORITY QUEUE**

F:\CDAC\CDAC\_Lab\Diwali Homework\Collection coding questions>java Que21\_QueueRemoveElement

Initial PriorityQueue: [5, 10, 30, 20, 15]

PriorityQueue after removing elements: [10, 20, 15]

PriorityQueue after removing all elements(clear() method use): []

F:\CDAC\CDAC\_Lab\Diwali Homework\Collection coding questions>\*/

**==============================================================**

**Que 22 – Write a Java program to count the number of key-value (size) mappings in a map**

**Ans🡪**

import java.util.\*;

class HashMapCountOfKeyValuePair22{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

HashMap<Integer, String> m1 = new HashMap<Integer, String>();

m1.put(1, "AAAAAA");

m1.put(2, "BBBBBB");

m1.put(3, "CCCCCC");

m1.put(4, "DDDDDD");

m1.put(5, "EEEEEE");

m1.put(6, "FFFFFF");

m1.put(7, "GGGGGG");

System.out.println("Elements in the hashmap are : ");

int count = 0;

Set<Map.Entry<Integer, String>> s1 = m1.entrySet();

for(Map.Entry<Integer, String> entry : s1){

System.out.println(entry);

count++;

}

System.out.println("Total key-value pairs in the hashmap are : "+count);

}

}

**==============================================================**

**Que 23 – Write a Java program to convert a priority queue to an array containing all of the elements of the queue**

**Ans🡪**

import java.util.\*;

class Collection23

{

public static void main(String[] args)

{

PriorityQueue<String> pq1 = new PriorityQueue<String>();

pq1.add("Red");

pq1.add("Green");

pq1.add("Black");

pq1.add("White");

System.out.println("Original Priority Queue: "+pq1);

List<String> array\_list = new ArrayList<String>(pq1);

System.out.println("Array containg all of the elements in the queue: "+array\_list);

}

}

**==============================================================**

**Que 24 – Write a Java program to check whether a map contains key-value mappings (empty) or not**

**Ans🡪**

**==============================================================**

**Que 25 – Write a Java program to get the value of a specified key in a map**

**Ans🡪**

import java.lang.\*;

import java.util.\*;

class HashMapEmpty24{

public static void main(String args[]){

HashMap<Integer, String> hm1 = new HashMap<Integer, String>();

hm1.put(1, "Red");

hm1.put(2, "Blue");

hm1.put(3, "Orange");

hm1.put(4, "Yellow");

hm1.put(5, "Pink");

hm1.put(6, "Black");

hm1.put(7, "Brown");

hm1.put(8, "White");

if(hm1.isEmpty()){

System.out.println("Hash map is empty");

}

else{

System.out.println("Hash map is not empty");

}

}

}