

ASSIGNMENT – 2.2

Subject: CSW2 (CSE 2141)

Name: Arpit Kumar Mohanty

Registration Number: 2341013237

Section: 23412G1

Branch: CSE

Q1. Create a generic class Pair with private member variables key and value. The class should include a parameterized constructor and provide getter and setter methods for these attributes. In the main class, create and add objects of the Pair class, then retrieve and print the key-value pairs.

Solution along with Output:



```
J Q1.java x J Q2.java J Q3.java J Q4.java J Q5.java J Q6.java J Q7.java J Q8.java J Q9.java J Q10.java
J Q1.java > Q1
2 class Pair<K, V> {
3     private K key;    private V value;
4     // Parameterized Constructor
5     public Pair(K key, V value) {
6         this.key = key;
7         this.value = value;
8     }
9
10    public K getKey() { return key; }
11    public V getValue() { return value; }
12    // Setters
13    public void setKey(K key) { this.key = key; }
14    public void setValue(V value) { this.value = value; }
15 }
16
17 public class Q1 {
18     Run | Debug
19     public static void main(String args[]) {
20         // Creating Pair objects with different types
21         Pair<Integer, String> p1 = new Pair<>(key:10, value:"One");
22         Pair<String, Double> p2 = new Pair<>(key:"Price", value:99.99);
23
24         // Retrieving and printing values
25         System.out.println("Pair 1 - Key: " + p1.getKey() + ", Value: " + p1.getValue());
26         System.out.println("Pair 2 - Key: " + p2.getKey() + ", Value: " + p2.getValue());
27     }
28 }
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS Filter

[Running] cd "a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\05-03-2025 (Chap-13) ASSIGNMENT"

Pair 1 - Key: 10, Value: One

Pair 2 - Key: Price, Value: 99.99

Q2. Write a Java program that includes a User class and an ArrayListUser class. The User class should have private fields for name and age, along with a parameterized constructor and getter/setter methods for these attributes. The ArrayListUser class should create an ArrayList of User objects. After adding user objects, it should retrieve and print their name and age. Additionally, the program should sort the users based on age and print the updated list of users using getter methods.

Solution:

```
J Q1.java J Q2.java X J Q3.java J Q4.java J Q5.java J Q6.java J Q7.java J Q8.java J
J Q2.java > Q2 > main(String[])
1  import java.util.*;
2  class User {
3      private String name;
4      private int age;
5      public User(String name, int age) {
6          this.name = name;
7          this.age = age;
8      }
9      public String getName() { return name; }
10     public int getAge() { return age; }
11     public void setName(String name) { this.name = name; }
12     public void setAge(int age) { this.age = age; }
13
14     public void display() {
15         System.out.println("Name: " + name + " | Age: " + age);
16     }
17 }
18 public class Q2 {
    Run | Debug
19     public static void main(String[] args) {
20         // Creating an ArrayList of User objects
21         ArrayList<User> arl = new ArrayList<>();
22
23         arl.add(new User(name:"Arpit Kumar", age:20));
24         arl.add(new User(name:"Manshik Kumar", age:22));
25         arl.add(new User(name:"Sourav Sahoo", age:19));
26
27         System.out.println(x:"Users before sorting:");
28         for (User user : arl) {
29             user.display();
30         }
31
32         // Sorting users by age
33         Collections.sort(arl, Comparator.comparingInt(User::getAge));
34
35         System.out.println(x:"\nUsers after sorting by age:");
36         for (User user : arl) {
37             user.display();
38         }
39     }
40 }
```

Output:

```
[Running] cd "a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-
Users before sorting:
Name: Arpit Kumar | Age: 20
Name: Manshik Kumar | Age: 22
Name: Sourav Sahoo | Age: 19

Users after sorting by age:
Name: Sourav Sahoo | Age: 19
Name: Arpit Kumar | Age: 20
Name: Manshik Kumar | Age: 22

[Done] exited with code=0 in 1.124 seconds
```

Q3. Write a Java program that includes a Car class and a CarApp class. The Car class should have private fields: modelNo (int), name (String), and stock (int). Define a parameterized constructor and override the compareTo method as public int compareTo(Car car) to enable sorting of cars based on the total stock available. In the CarApp class, create an ArrayList of Car objects, sort them, and print the updated sorted list. Example of a sorted list of Car objects:

2013 Creta 10

2020 MG 13

2018 Kia 20

2017 Audi 45

2015 BMW 55

Solution along with Output:

```
J Q1.java J Q2.java J Q3.java X J Q4.java J Q5.java J Q6.java J Q7.java J Q8.java J Q9.java
J Q3.java > Q3
1  import java.util.*;
2  class Car implements Comparable<Car> {
3      private int modelNo;    private String name;    private int stock;
4
5      public Car(int modelNo, String name, int stock) {
6          this.modelNo = modelNo;
7          this.name = name;
8          this.stock = stock;
9      }
10
11     public int getStock() {
12         return stock;
13     }
14
15     // Override compareTo method to sort based on stock
16     @Override
17     public int compareTo(Car car) {
18         return Integer.compare(this.stock, car.stock);
19     }
20
21     // Override toString for easy printing
22     @Override
23     public String toString() {
24         return modelNo + " " + name + " " + stock;
25     }
26 }
27
```

```

28 public class Q3 {
    Run | Debug
29     public static void main(String[] args) {
30         ArrayList<Car> cars = new ArrayList<>();
31
32         // Adding Car objects to the ArrayList
33         cars.add(new Car(modelNo:2015, name:"BMW", stock:55));
34         cars.add(new Car(modelNo:2017, name:"Audi", stock:45));
35         cars.add(new Car(modelNo:2018, name:"Kia", stock:20));
36         cars.add(new Car(modelNo:2020, name:"MG", stock:13));
37         cars.add(new Car(modelNo:2013, name:"Creta", stock:10));
38         // Sorting cars based on stock
39         Collections.sort(cars);
40
41         // Printing sorted car list
42         System.out.println(x:"Sorted list of cars:");
43         for (Car car : cars) {
44             System.out.println(car);
45         }
46     }
47 }

```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

[Running] cd "a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\05-03-2025 (
Sorted list of cars:
2013 Creta 10
2020 MG 13
2018 Kia 20
2017 Audi 45
2015 BMW 55

[Done] exited with code=0 in 1.186 seconds

```

Q4. Create a Student class with member variables name, age, and mark, along with the necessary getter and setter methods. Implement a LinkedList of Student objects and perform the following operations: (a) Display the list of students. (b) Prompt the user to enter a Student object and check its existence in the list. Specify whether the search is based on reference comparison or content comparison using the contains method. (c) Remove a specified Student object from the list. (d) Count the number of Student objects present in the list. (e) Override the equals method to compare two Student objects based on their values rather than references

Solution along with Output:

```
Q1.java Q2.java Q3.java Q4.java X Q5.java Q6.java Q7.java Q8.java Q9.java Q10.java Q11.java
Q4.java > Q4
1  import java.util.*;
2  class Student {
3      String name;    int age;    double mark;
4
5      public Student(String name, int age, double mark) {
6          this.name = name;
7          this.age = age;
8          this.mark = mark;
9      }
10
11     @Override
12     public boolean equals(Object obj) {
13         if (this == obj) return true;
14         if (obj == null || getClass() != obj.getClass()) return false;
15         Student student = (Student) obj;
16         return age == student.age && Double.compare(student.mark, mark) == 0 && name.equals(student.name);
17     }
18
19     @Override
20     public int hashCode() {
21         return Objects.hash(name, age, mark);
22     }
23
24     @Override
25     public String toString() {
26         return "Student{name='" + name + "', age=" + age + ", mark=" + mark + "}";
27     }
28 }
29
30 public class Q4 {
    Run | Debug
31     public static void main(String[] args) {
32         LinkedList<Student> students = new LinkedList<>();
33         Scanner sc = new Scanner(System.in);
34
35         students.add(new Student(name:"Alice", age:20, mark:85.5));
36         students.add(new Student(name:"Bob", age:22, mark:90.0));
37         students.add(new Student(name:"Charlie", age:19, mark:78.0));
38
39         // (a) Display the list of students
40         System.out.println(x:"List of students:");
41         for (Student s : students) {
42             System.out.println(s);
43         }
44     }
45 }
```

```
J Q1.java J Q2.java J Q3.java J Q4.java X J Q5.java J Q6.java J Q7.java J Q8.java J Q9.java J Q10.java
J Q4.java > Q4
30 public class Q4 {
31     public static void main(String[] args) {
44
45         // (b) Check for existence of a Student object
46         System.out.println(x:"\nEnter student details to search (name, age, mark):");
47         String name = sc.next();
48         int age = sc.nextInt();
49         double mark = sc.nextDouble();
50         Student searchStudent = new Student(name, age, mark);
51         if (students.contains(searchStudent)) {
52             System.out.println(x:"Student found (content comparison).\n");
53         } else {
54             System.out.println(x:"Student not found.\n");
55         }
56
57         // (c) Remove a Student object
58         System.out.println(x:"Enter student details to remove (name, age, mark):");
59         name = sc.next();
60         age = sc.nextInt();
61         mark = sc.nextDouble();
62         Student removeStudent = new Student(name, age, mark);
63         if (students.remove(removeStudent)) {
64             System.out.println(x:"Student removed successfully.\n");
65         } else {
66             System.out.println(x:"Student not found for removal.\n");
67         }
68
69         // (d) Count the number of students
70         System.out.println("Total number of students: " + students.size());
71         sc.close();
72     }
73 }
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

[Running] cd "a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\05-03-2025 (Chap-13) ASSIGNMENT 2.1

List of students:

Student{name='Alice', age=20, mark=85.5}

Student{name='Bob', age=22, mark=90.0}

Student{name='Charlie', age=19, mark=78.0}

Enter student details to search (name, age, mark):

Q5. Create a Book class with member variables id, name, author, and quantity to store details of each issued book. The Book class should include a parameterized constructor. Design a Library class that creates a HashMap of books, where the key is an Integer (representing the book ID) and the value is a Book object. Instantiate at least two Book objects and display the collection of books stored in the HashMap. Use appropriate methods of the HashMap class to perform the following operations: (a) Check if a particular book name is present on the map. (b) Remove a book entry by deleting the value associated with a specific

key.

Solution:

```
J Q1.java J Q2.java J Q3.java J Q4.java J Q5.java X J Q6.java J Q7.java J Q8.java J Q9.java J Q10.java J Q11.java
J Q5.java > Book
1 import java.util.*;
2 class Book {
3     int id, quantity;
4     String name, author;
5
6     public Book(int id, String name, String author, int quantity) {
7         this.id = id;
8         this.name = name;
9         this.author = author;
10        this.quantity = quantity;
11    }
12
13    @Override
14    public String toString() {
15        return "Book ID: " + id + ", Name: " + name + ", Author: " + author + ", Quantity: " + quantity;
16    }
17 }
18
19 public class Q5 {
20     Run | Debug
21     public static void main(String[] args) {
22         HashMap<Integer, Book> bookMap = new HashMap<>();
23         Scanner sc = new Scanner(System.in);
24
25         // Adding books to the HashMap
26         bookMap.put(key:101, new Book(id:001, name:"Basics of Human", author:"James", quantity:5));
27         bookMap.put(key:102, new Book(id:060, name:"Python Basics", author:"Hecks", quantity:3));
28
29         // Displaying all books
30         System.out.println(x:"Library Collection:");
31         for (Map.Entry<Integer, Book> entry : bookMap.entrySet()) {
32             System.out.println(entry.getValue());
33         }
34
35         // (a) Check if a particular book name is present in the map using a simple loop
36         System.out.println(x:"\nEnter book name to search:");
37         String searchName = sc.nextLine();
38         boolean found = false;
39         for (Book book : bookMap.values()) {
40             if (book.name.equalsIgnoreCase(searchName)) {
41                 found = true;
42                 break;
43             }
44         }
45     }
46 }
```

```
J Q1.java J Q2.java J Q3.java J Q4.java J Q5.java X J Q6.java J Q7.java J Q8.java J Q9.java J Q10.java J Q11.java
J Q5.java > Q5 > main(String[])
19 public class Q5 {
20     public static void main(String[] args) {
21         // (a) Display the elements of the TreeSet
22         System.out.println(found ? "Book is available in the library." : "Book is not available.");
23     }
24     // (b) Remove a book entry by deleting the value associated with a specific key
25     System.out.println(x: "\nEnter book ID to remove:");
26     int removeId = sc.nextInt();
27     if (bookMap.containsKey(removeId)) {
28         bookMap.remove(removeId);
29         System.out.println(x: "Book removed successfully.");
30     } else {
31         System.out.println(x: "Book ID not found.");
32     }
33     // Display updated book collection
34     System.out.println(x: "\nUpdated Library Collection:");
35     for (Map.Entry<Integer, Book> entry : bookMap.entrySet()) {
36         System.out.println(entry.getValue());
37     }
38     sc.close();
39 }
40 }
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

[Done] exited with code=1 in 78.794 seconds

[Running] cd "a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\05-03-2025 (Chap-13) ASSIGNMENT 2.1 & 2.2\A9

Library Collection:

Book ID: 1, Name: Basics of Human, Author: James, Quantity: 5

Book ID: 48, Name: Python Basics, Author: Hecks, Quantity: 3

Enter book name to search:

Q6. Write a program to create a TreeSet of Integer type and perform the following operations: (a) Display the elements of the TreeSet. (b) Prompt the user to enter a number and check whether the number is present in the TreeSet. (c) Remove a specified element from the TreeSet

Solution:



```
1 import java.util.*;
2 public class Q6 {
3     public static void main(String[] args) {
4         Scanner sc = new Scanner(System.in);
5         TreeSet<Integer> treeSet = new TreeSet<>();
6
7         for (int i = 10; i <= 50; i += 10) {
8             treeSet.add(i);
9         }
10
11         // (a) Display the elements of the TreeSet
12         System.out.println("TreeSet elements: " + treeSet);
13
14         // (b) Check if a user-input number is present in the TreeSet
15         System.out.print("Enter a number to check: ");
16         int numberToCheck = sc.nextInt();
17         System.out.println(numberToCheck + (treeSet.contains(numberToCheck) ? " is present in the TreeSet." : " is not present in the TreeSet."));
18
19         // (c) Remove a specified element from the TreeSet
20         System.out.print("Enter a number to remove: ");
21         int numberToRemove = sc.nextInt();
22         System.out.println(numberToRemove + (treeSet.remove(numberToRemove) ? " has been removed from the TreeSet." : " was not found in the TreeSet."));
23
24         // Display updated TreeSet
25         System.out.println("Updated TreeSet: " + treeSet);
26         sc.close();
27     }
28 }
```

[Done] exited with code=1 in 45.976 seconds

[Running] cd "a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\05-03-2025 (Chap-13) ASSIGNMENT 2.1 & 2.2\ASSIGNMENT 2.2\" && javac Q6.java && java Q6

TreeSet elements: [10, 20, 30, 40, 50]

Enter a number to check:

Q7. Write a Java program that includes a class Address with member variables plotNo, at, and post. The class should define a parameterized constructor to initialize these attributes. Create a TreeMap, where the key is the name of a person (String), and the value is an Address object. Insert the required key-value pairs into the TreeMap and use an Iterator to display the entries.

Solution:

```
J Q1.java J Q2.java J Q3.java J Q4.java J Q5.java J Q6.java J Q7.java X J Q8.java J Q9.java J Q10.java J Q11.java
J Q7.java > Q7 > main(String[])
1  import java.util.*;
2  class Address {
3      int plotNo;    String at;    String post;
4
5      public Address(int plotNo, String at, String post) {
6          this.plotNo = plotNo;
7          this.at = at;
8          this.post = post;
9      }
10
11     @Override
12     public String toString() {
13         return "Plot No: " + plotNo + ", At: " + at + ", Post: " + post;
14     }
15 }
16
17 public class Q7 {
18     Run | Debug
19     public static void main(String[] args) {
20         TreeMap<String, Address> addressMap = new TreeMap<>();
21
22         // Adding entries to the TreeMap
23         addressMap.put(key:"Arpit", new Address(plotNo:159, at:"ABC Lane", post:"Delhi"));
24         addressMap.put(key:"Bhuvan", new Address(plotNo:202, at:"BCD Lane", post:"BBSR"));
25         addressMap.put(key:"Chinmayee", new Address(plotNo:893, at:"EDC Lane", post:"Puri"));
26
27         // Using an iterator to display the entries
28         Iterator<Map.Entry<String, Address>> iterator = addressMap.entrySet().iterator();
29         while (iterator.hasNext()) {
30             Map.Entry<String, Address> entry = iterator.next();
31             System.out.println("Name: " + entry.getKey() + ", Address: " + entry.getValue());
32         }
33     }
34 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[Running] cd "a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\05-03-2025 (Chap-13) ASSIGNMENT 2.1 & 2.2\ASSIGNMENT 2

Name: Arpit, Address: Plot No: 159, At: ABC Lane, Post: Delhi

Name: Bhuvan, Address: Plot No: 202, At: BCD Lane, Post: BBSR

Name: Chinmayee, Address: Plot No: 893, At: EDC Lane, Post: Puri

Q8. Write a Java program to determine whether two given strings are anagrams. An anagram is a word or phrase formed by rearranging the letters of another word or phrase. Declare two strings, str1 and str2, and initialize them with values. Create a HashMap to store the character frequencies of one string. Use the methods containsKey(), put(), and get() to compare both strings and verify if they are anagrams.

Solution:




```
1 import java.util.*;
2 public class Q8 {
3     public static boolean areAnagrams(String str1, String str2) {
4         if (str1.length() != str2.length()) return false;
5
6         HashMap<Character, Integer> charCountMap = new HashMap<>();
7
8         for (char c : str1.toCharArray())
9             charCountMap.put(c, charCountMap.getOrDefault(c, 0) + 1);
10
11        for (char c : str2.toCharArray()) {
12            if (!charCountMap.containsKey(c) || charCountMap.get(c) == 0) return false;
13            charCountMap.put(c, charCountMap.get(c) - 1);
14        }
15        return true;
16    }
17
18    public static void main(String[] args) {
19        String str1 = "listen", str2 = "silent";
20        System.out.println(str1 + " and " + str2 + (areAnagrams(str1, str2) ? " are " : " are not ") + "anagrams.");
21    }
22 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[Running] cd "a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\05-03-2025 (Chap-13) ASSIGNMENT 2.1 & 2.2\ASSIGNMENT 2.2\" && ;
listen and silent are anagrams.

Q9. Given an array of integers, write a Java program to identify and print the repeating integers using a HashSet.

Solution:



```
1 import java.util.*;
2 public class Q9 {
3     public static void main(String[] args) {
4         int[] arr = {1, 2, 3, 4, 5, 2, 5, 6, 7, 8, 6};
5         HashSet<Integer> seen = new HashSet<>();
6         HashSet<Integer> duplicates = new HashSet<>();
7
8         for (int i = 0; i < arr.length; i++) {
9             if (!seen.add(arr[i])) {
10                 duplicates.add(arr[i]);
11             }
12         }
13         System.out.println("Repeating integers: " + duplicates);
14     }
15 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

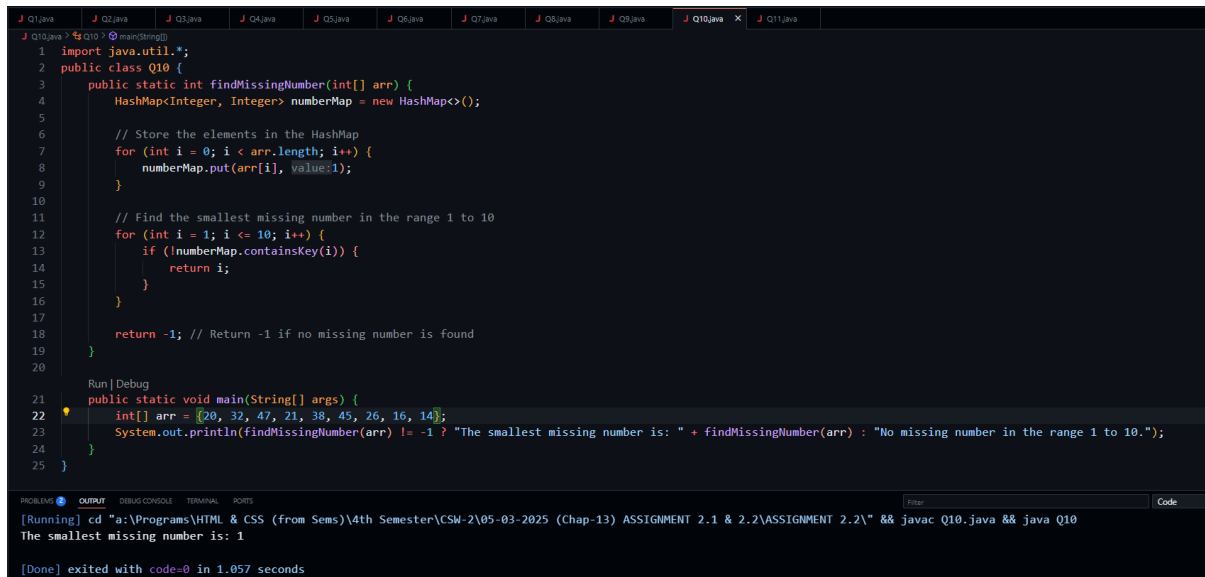
[Running] cd "a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\05-03-2025 (Chap-13) ASSIGNMENT 2.1 & 2.2\ASSIGNMENT 2.2"
Repeating integers: [2, 5, 6]

[Done] exited with code=0 in 0.992 seconds

Q10. Given an unsorted array of integers ranging from 1 to 10, write a program to find the smallest positive number missing in the array. Use a HashMap to keep track of the elements and identify the missing

number.

Solution:



```
1 import java.util.*;
2 public class Q10 {
3     public static int findMissingNumber(int[] arr) {
4         HashMap<Integer, Integer> numberMap = new HashMap<>();
5
6         // Store the elements in the HashMap
7         for (int i = 0; i < arr.length; i++) {
8             numberMap.put(arr[i], value:1);
9         }
10
11        // Find the smallest missing number in the range 1 to 10
12        for (int i = 1; i <= 10; i++) {
13            if (!numberMap.containsKey(i)) {
14                return i;
15            }
16        }
17
18        return -1; // Return -1 if no missing number is found
19    }
20
21    Run | Debug
22    public static void main(String[] args) {
23        int[] arr = {20, 32, 47, 21, 38, 45, 26, 16, 14};
24        System.out.println(findMissingNumber(arr) != -1 ? "The smallest missing number is: " + findMissingNumber(arr) : "No missing number in the range 1 to 10.");
25    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[Running] cd "a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\05-03-2025 (Chap-13) ASSIGNMENT 2.1 & 2.2\ASSIGNMENT 2.2\" && javac Q10.java && java Q10

The smallest missing number is: 1

[Done] exited with code=0 in 1.057 seconds

Q11. Declare an array of integers: `int[] arr = {1, 2, 10, 8, 7, 3, 4, 6, 5, 9};`. Then, create a min-heap using the `PriorityQueue` class to store the elements from the array. Finally, dequeue the elements from the `PriorityQueue` using the appropriate methods and print them

Solution:



```
J Q1.java > Q11 > main(String[])
1  import java.util.*;
2  public class Q11 {
    Run | Debug
3      public static void main(String[] args) {
4          int[] arr = {7, 3, 9, 1, 5, 8, 2, 10, 6, 4};
5
6          // Create a min-heap (PriorityQueue)
7          PriorityQueue<Integer> minHeap = new PriorityQueue<>();
8
9          // Insert elements into the min-heap
10         for (int i = 0; i < arr.length; i++) {
11             minHeap.add(arr[i]);
12         }
13
14         // Dequeue elements and print them
15         System.out.println(x:"Elements in sorted order:");
16         while (!minHeap.isEmpty()) {
17             System.out.print(minHeap.poll() + " ");
18         }
19     }
20 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[Running] cd "a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\05-03-2025 (Chap-13) ASSIGNMENT 2.1 & 2.2\ASSIGNMENT 2.2\"
Elements in sorted order:
1 2 3 4 5 6 7 8 9 10
[Done] exited with code=0 in 1.08 seconds