**Assisted Practice: 5.7 Merge Sort**

This section will guide you to:

* Create a Java class in your IDE
* Implement the merge sort algorithm in a predefined array
* Check if the element is available in the predefined array

This lab has three subsections, namely:

5.7.1 Writing a program in Java implementing the merge sort algorithm

5.7.2 Executing the program to verify the execution of the merge sort algorithm

5.7.3 Pushing the code to your GitHub repositories

**Step 5.7.1:** Write a program in Java implementing the merge sort algorithm

There are two ways you can perform this step; you can create a new Java project, or you can create a new Java class in the existing project. It is preferable to create a new Java class in the existing project but feel free to explore the first option. The steps mentioned below will work once you create a project in Java.

* Open Eclipse
* *[Right click]* on the **src** folder of the project
* Select *New* -> *Java Class* -> Enter the filename (follow camelCasing)
* Execute the code below resolving the warning and errors due compatibility-related issues

/\* Java program for Merge Sort \*/  
class MergeSort  
{  
   
 void merge(int arr[], int l, int m, int r)  
 {  
   
 int n1 = m - l + 1;  
 int n2 = r - m;  
  
 /\* Create temp arrays \*/  
 int L[] = new int [n1];  
 int R[] = new int [n2];  
  
 /\*Copy data to temp arrays\*/  
 for (int i=0; i<n1; ++i)  
 L[i] = arr[l + i];  
 for (int j=0; j<n2; ++j)  
 R[j] = arr[m + 1+ j];  
  
  
  
 int i = 0, j = 0;  
  
 int k = l;  
 while (i < n1 && j < n2)  
 {  
 if (L[i] <= R[j])  
 {  
 arr[k] = L[i];  
 i++;  
 }  
 else  
 {  
 arr[k] = R[j];  
 j++;  
 }  
 k++;  
 }  
 while (i < n1)  
 {  
 arr[k] = L[i];  
 i++;  
 k++;  
 }  
  
   
 while (j < n2)  
 {  
 arr[k] = R[j];  
 j++;  
 k++;  
 }  
 }  
 void sort(int arr[], int l, int r)  
 {  
 if (l < r)  
 {  
   
 int m = (l+r)/2;  
  
   
 sort(arr, l, m);  
 sort(arr , m+1, r);  
  
   
 merge(arr, l, m, r);  
 }  
 }  
  
 static void printArray(int arr[])  
 {  
 int n = arr.length;  
 for (int i=0; i<n; ++i)  
 System.*out*.print(arr[i] + " ");  
 System.*out*.println();  
 }  
  
 // Driver method  
 public static void main(String args[])  
 {  
 int arr[] = {12, 11, 13, 5, 6, 7};  
  
 System.*out*.println("Given Array");  
 *printArray*(arr);  
  
 MergeSort ob = new MergeSort();  
 ob.sort(arr, 0, arr.length-1);  
  
 System.*out*.println("\nSorted array");  
 *printArray*(arr);  
 }  
}

**Step 5.7.2:** Execute the program to verify the execution of the merge sort algorithm

Before you execute the program, check for syntactical corrections. If no errors are found, follow the steps mentioned below:

* ***[Right click]*** in the program space
* Select *Run* ***‘MergeSort.main()’***



**Step 5.7.3:** Pushing the code to your GitHub repositories

* Open your command prompt and navigate to the folder where you have created your files.

**cd <folder path>**

* Initialize your repository using the following command:

**git init**

* Add all the files to your git repository using the following command:

**git add .**

* Commit the changes using the following command:

**git commit . -m “Changes have been committed.”**

* Push the files to the folder you initially created using the following command:

**git push -u origin master**