Bardia Ardakanian

9831072

Q1:

1.A class is a user defined blueprint or prototype from which objects are created.  It represents the set of properties or methods that are common to all objects of one type.

2. A **function is** a piece of code that **is** called by name. ... A **method is** a piece of code that **is** called by a name that **is** associated with an object. In most respects it **is** identical to a **function** except for two key **differences**: A **method is** implicitly passed the object on which it was called.

3. The access modifiers in Java specifies the accessibility or scope of a field, method, constructor, or class. We can change the access level of fields, constructors, methods, and class by applying the access modifier on it.

There are four types of Java access modifiers:

1. **Private**: The access level of a private modifier is only within the class. It cannot be accessed from outside the class.
2. **Default**: The access level of a default modifier is only within the package. It cannot be accessed from outside the package. If you do not specify any access level, it will be the default.
3. **Protected**: The access level of a protected modifier is within the package and outside the package through child class. If you do not make the child class, it cannot be accessed from outside the package.
4. **Public**: The access level of a public modifier is everywhere. It can be accessed from within the class, outside the class, within the package and outside the package

There are many non-access modifiers, such as static, abstract, synchronized, native, volatile, transient, etc. Here, we are going to learn the access modifiers only.

Q2:

1. false. //The access level of a protected modifier is within the package

2. true.

3. true.

4. true.

5. false. //primitive data type of decimal numbers are double

6. true. // **Java** is **statically**-**typed**, so it expects its variables to be declared before they can be assigned values.

7. true.

Q3:

package com.company;  
*/\*\*  
 \* this method uses something like merge sort and sort array from highest number to lowest  
 \** ***@author*** *bardia ardakanian 9831072  
 \*/*class Main  
{  
 static int part(int[] arr, int low, int high)  
 {  
 int pivot = arr[high], i = low;  
 {  
 for (; low < high; low++) {  
 if (arr[i] > pivot) {  
 int temp = arr[i];  
 arr[i] = arr[low];  
 arr[low] = temp;  
 i++;  
 }  
 }  
 int temp = arr[i];  
 arr[i] = arr[high];  
 arr[high] = temp;  
 }  
 return i;  
 }  
  
 static void sort(int[] arr, int low, int high)  
 {  
 if (low < high)  
 {  
 int p = *part*(arr, low, high);  
 *sort*(arr, low, p-1);  
 *sort*(arr, p+1, high);  
 }  
 }  
  
 static void print(int... arr)  
 {  
 for (int i : arr)  
 System.*out*.print(i + " ");  
 System.*out*.println();  
 }  
  
 public static void main(String args[])  
 {  
 int[] a = {10, 7, 8, 9, 1, 5};  
 int n = a.length;  
 *sort*(a, 0, n-1);  
 *print*(a);  
 }  
}

this function uses recursive sort and it is very similar to merge sort.