

1/ Ans: Array -

```
// Using Array
String[] friendsArray = new String[3]; 4 usages
friendsArray[0] = "Alice";
friendsArray[1] = "Bob";
friendsArray[2] = "Charlie";

System.out.println("\nArray contents:");
for (String friend : friendsArray) {
    System.out.println(friend);
}
```

ArrayList -

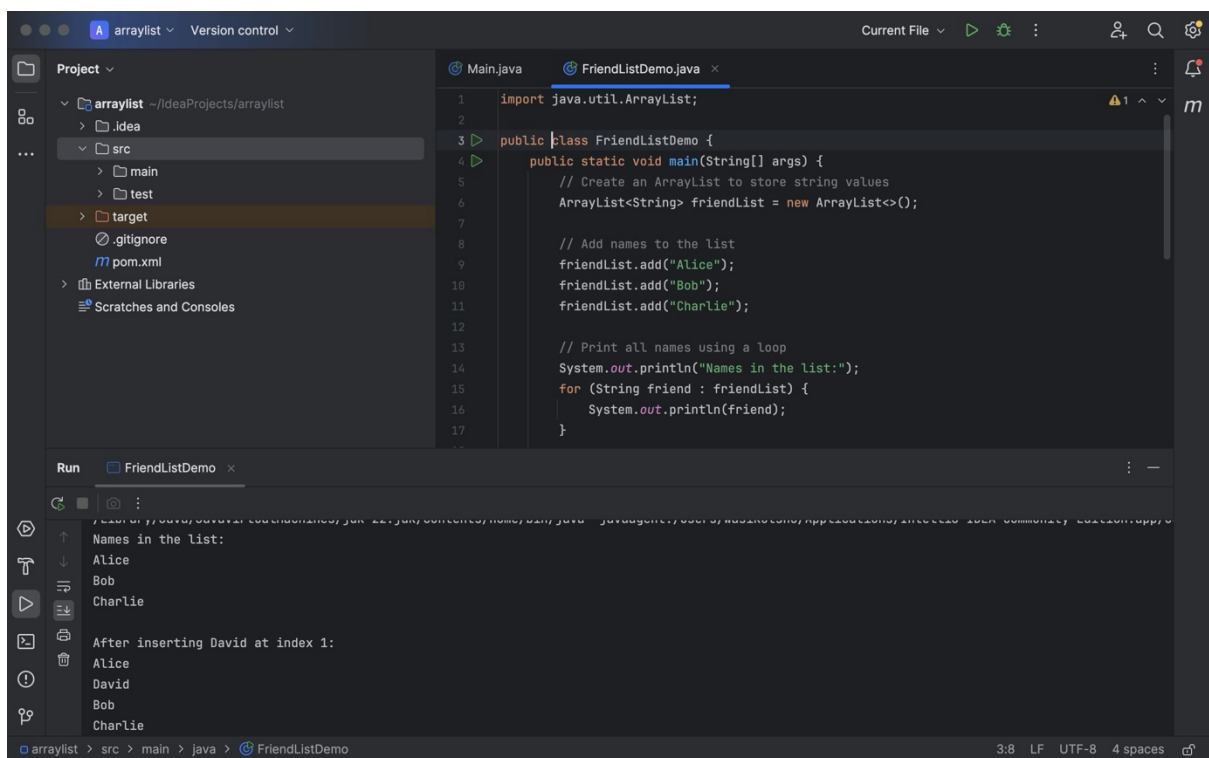
```
// Using ArrayList
ArrayList<String> friendsList = new ArrayList<>(); 1 usage
friendsList.add("Alice");
friendsList.add("Bob");
friendsList.add("Charlie");
friendsList.add(1, "David"); // Insert "David" at index 1
friendsList.remove("Bob"); // Remove "Bob" by value

System.out.println("ArrayList contents:");
for (String friend : friendsList) {
    System.out.println(friend);
}
```

## Difference between Array and ArrayList :

- a) ArrayLists can be dynamically resized, whereas arrays have a fixed size.
- b) ArrayLists can only hold objects; arrays can store primitive types as well.
- c) Arrays lack type safety; array lists employ generics to guarantee that only certain types are saved.
- d) An array list offers more ways to add, remove, and work with elements.
- e) When working with primitive types, arrays typically perform better and incur less overhead than ArrayLists.
- f) ArrayLists are a component of the Java Collections Framework, but arrays have a separate syntax and are a part of the core Java language.

2/ Ans:



The screenshot shows an IDE with a project named 'arraylist'. The 'src' directory contains 'main' and 'test' folders. The 'main' folder is selected, showing 'FriendListDemo.java'. The code in 'FriendListDemo.java' is as follows:

```
1 import java.util.ArrayList;
2
3 public class FriendListDemo {
4     public static void main(String[] args) {
5         // Create an ArrayList to store string values
6         ArrayList<String> friendList = new ArrayList<>();
7
8         // Add names to the list
9         friendList.add("Alice");
10        friendList.add("Bob");
11        friendList.add("Charlie");
12
13        // Print all names using a loop
14        System.out.println("Names in the list:");
15        for (String friend : friendList) {
16            System.out.println(friend);
17        }
18    }
19 }
```

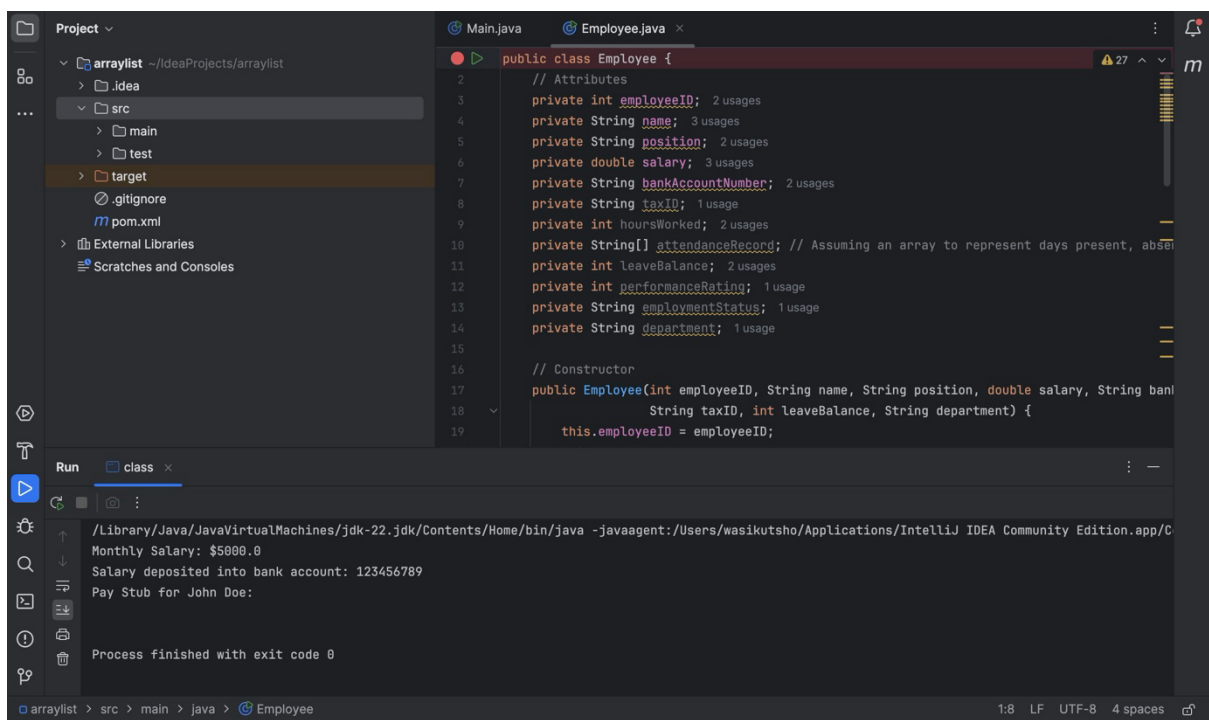
The 'Run' button is clicked, and the output is displayed in the 'Run' window:

```
Names in the list:
Alice
Bob
Charlie

After inserting David at index 1:
Alice
David
Bob
Charlie
```

The status bar at the bottom indicates the file is 'arraylist > src > main > java > FriendListDemo' with a line length of 3:8, LF, UTF-8, and 4 spaces.

3/ Ans:



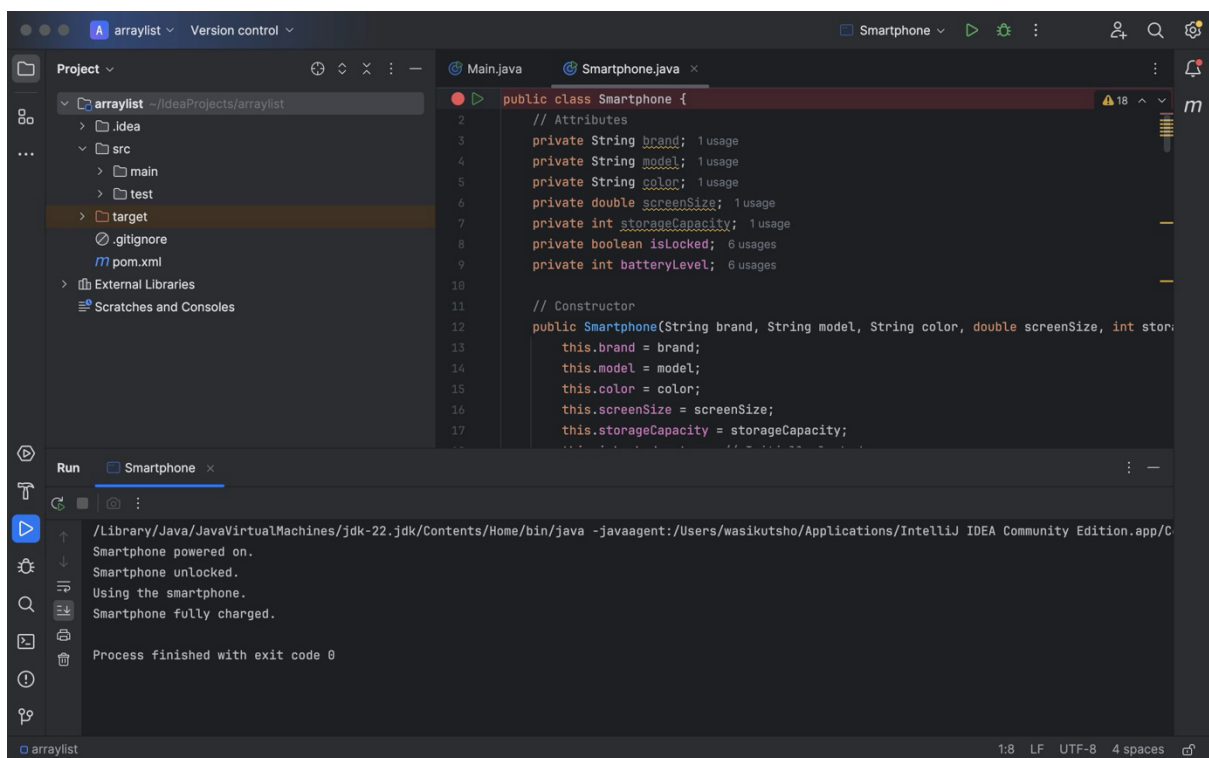
The screenshot shows the IntelliJ IDEA interface with the `Employee.java` file open. The file contains a `public class Employee` with various attributes and a constructor. The `Run` button is clicked, and the output is displayed in the `Run` tab.

```
public class Employee {  
    // Attributes  
    private int employeeID; 2 usages  
    private String name; 3 usages  
    private String position; 2 usages  
    private double salary; 3 usages  
    private String bankAccountNumber; 2 usages  
    private String taxID; 1 usage  
    private int hoursWorked; 2 usages  
    private String[] attendanceRecord; // Assuming an array to represent days present, absent  
    private int leaveBalance; 2 usages  
    private int performanceRating; 1 usage  
    private String employmentStatus; 1 usage  
    private String department; 1 usage  
  
    // Constructor  
    public Employee(int employeeID, String name, String position, double salary, String bankAccountNumber, String taxID, int leaveBalance, String department) {  
        this.employeeID = employeeID;  
    }  
}
```

Run class x

```
/Library/Java/JavaVirtualMachines/jdk-22.jdk/Contents/Home/bin/java -javaagent:/Users/wasikutsho/Applications/IntelliJ IDEA Community Edition.app/Contents/idea-ic-22.2.20230920/idea-ic-22.2.20230920.jar  
Monthly Salary: $5000.0  
Salary deposited into bank account: 123456789  
Pay Stub for John Doe:  
  
Process finished with exit code 0
```

4/Ans:



The screenshot shows the IntelliJ IDEA interface with the `Smartphone.java` file open. The file contains a `public class Smartphone` with various attributes and a constructor. The `Run` button is clicked, and the output is displayed in the `Run` tab.

```
public class Smartphone {  
    // Attributes  
    private String brand; 1 usage  
    private String model; 1 usage  
    private String color; 1 usage  
    private double screenSize; 1 usage  
    private int storageCapacity; 1 usage  
    private boolean isLocked; 6 usages  
    private int batteryLevel; 6 usages  
  
    // Constructor  
    public Smartphone(String brand, String model, String color, double screenSize, int storageCapacity, boolean isLocked, int batteryLevel) {  
        this.brand = brand;  
        this.model = model;  
        this.color = color;  
        this.screenSize = screenSize;  
        this.storageCapacity = storageCapacity;  
        this.isLocked = isLocked;  
        this.batteryLevel = batteryLevel;  
    }  
}
```

Run Smartphone x

```
/Library/Java/JavaVirtualMachines/jdk-22.jdk/Contents/Home/bin/java -javaagent:/Users/wasikutsho/Applications/IntelliJ IDEA Community Edition.app/Contents/idea-ic-22.2.20230920/idea-ic-22.2.20230920.jar  
Smartphone powered on.  
Smartphone unlocked.  
Using the smartphone.  
Smartphone fully charged.  
  
Process finished with exit code 0
```