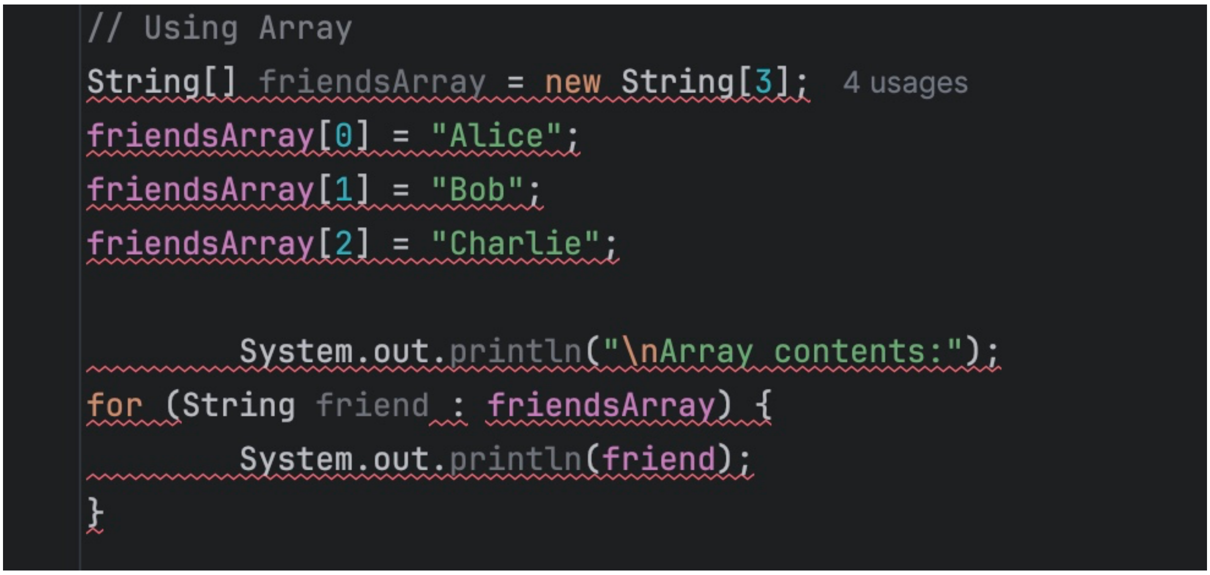
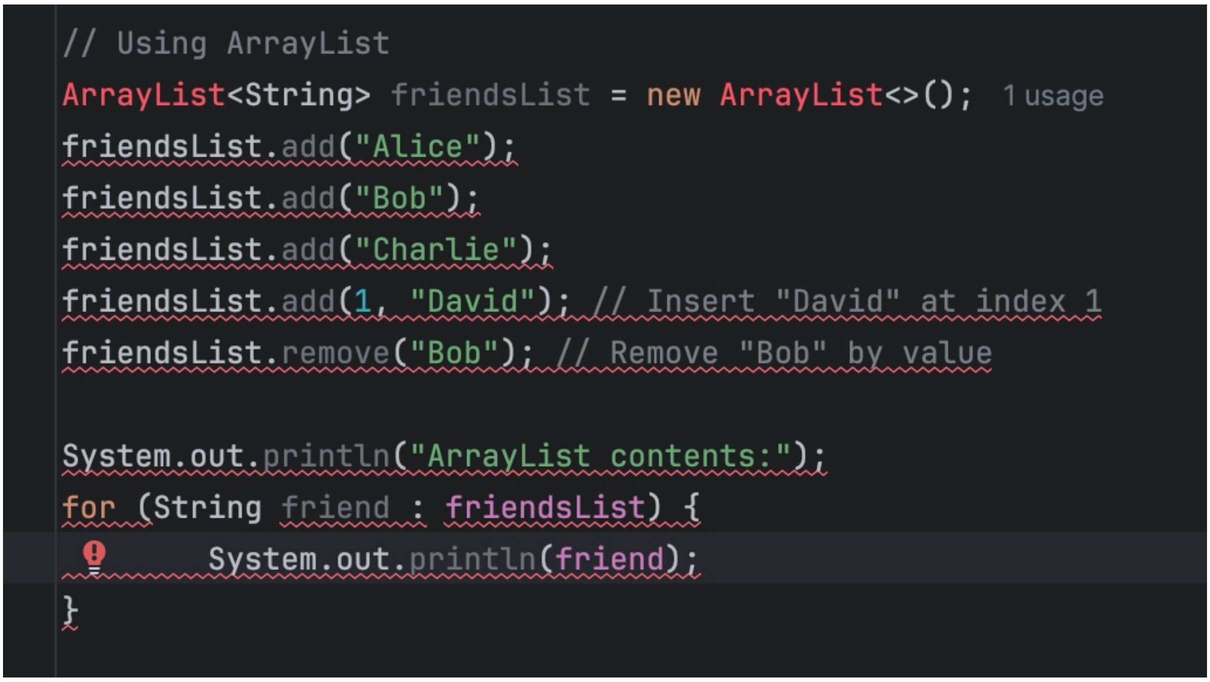
1/ Ans: Array -



```
// 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 -

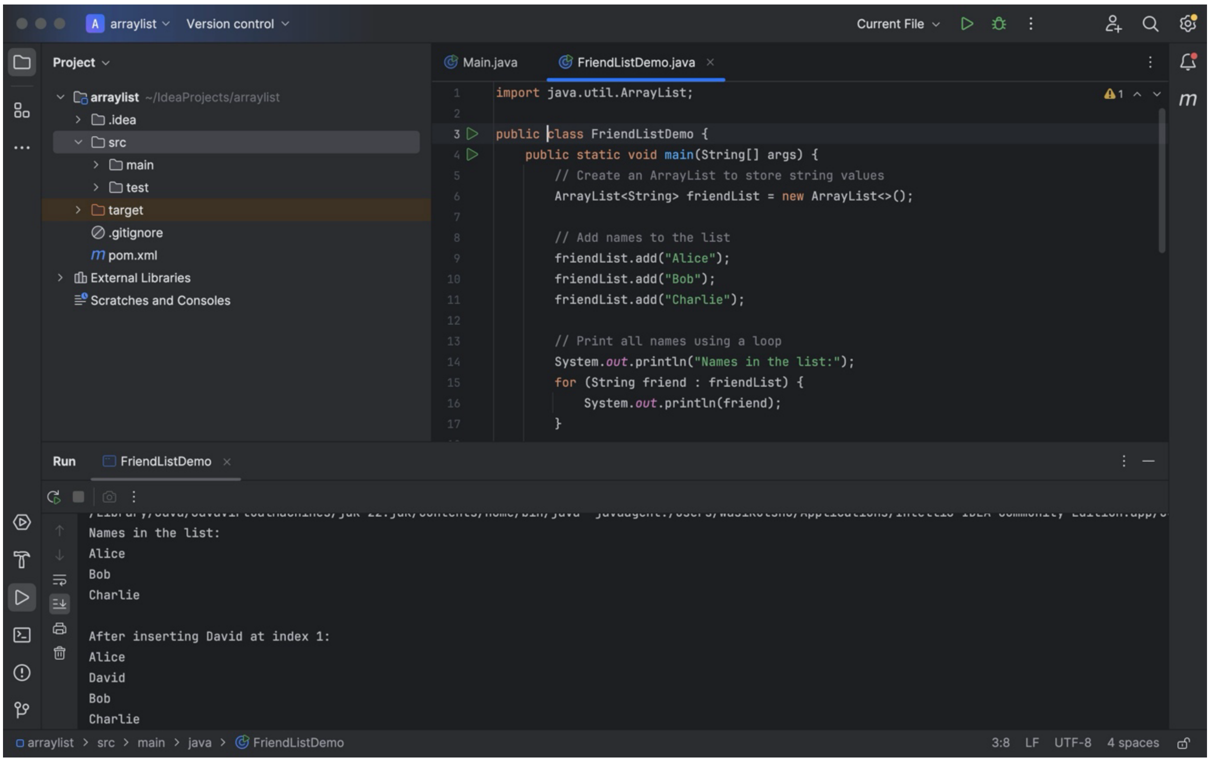


```
// Using ArrayList
ArrayList<String> friendsList = new ArrayList<>();  1 usage
friendsList.add("Alice");
friendsList.add("Bob");
friendsList.add("Charlie");
friendsList.add(1, "David"); // Insert "David" at index 1
friendsList.remove("Bob"); // Remove "Bob" by value

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.

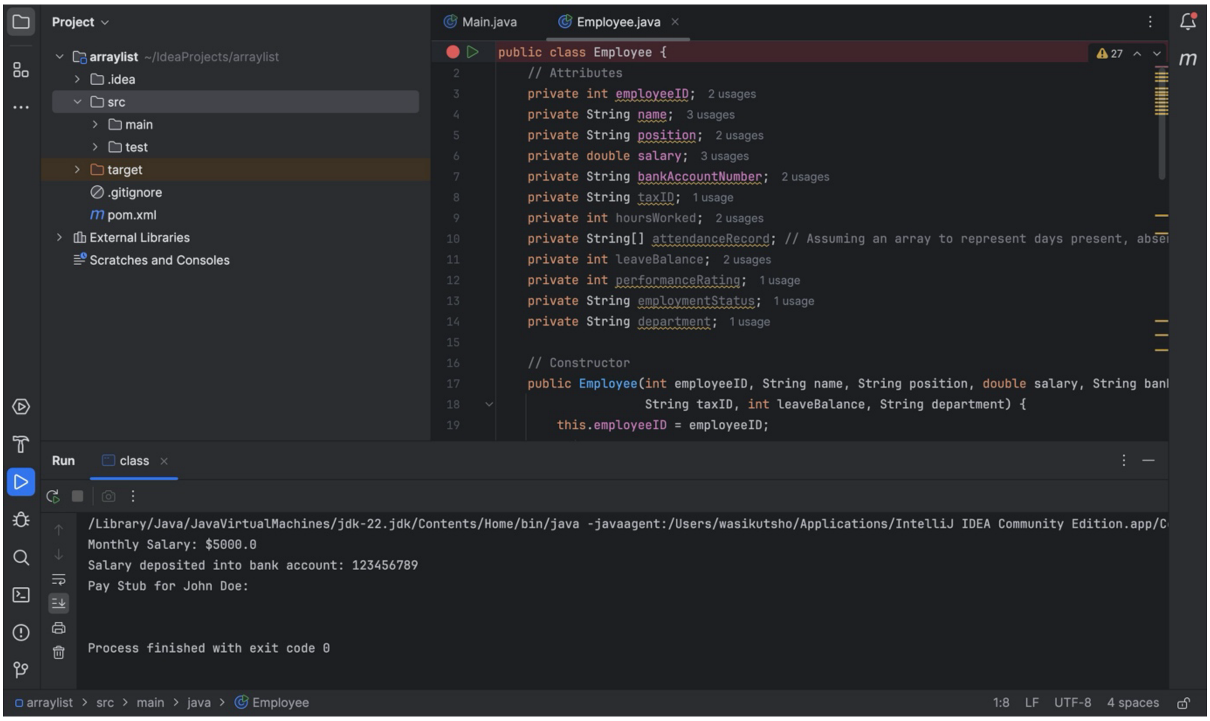
2/ Ans:



```java
import java.util.ArrayList;

public class FriendListDemo {
    public static void main(String[] args) {
        // Create an ArrayList to store string values
        ArrayList<String> friendList = new ArrayList<>();

        // Add names to the list
        friendList.add("Alice");
        friendList.add("Bob");
        friendList.add("Charlie");

        // Print all names using a loop
        System.out.println("Names in the list:");
        for (String friend : friendList) {
            System.out.println(friend);
        }
```

```
Names in the list:
Alice
Bob
Charlie

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

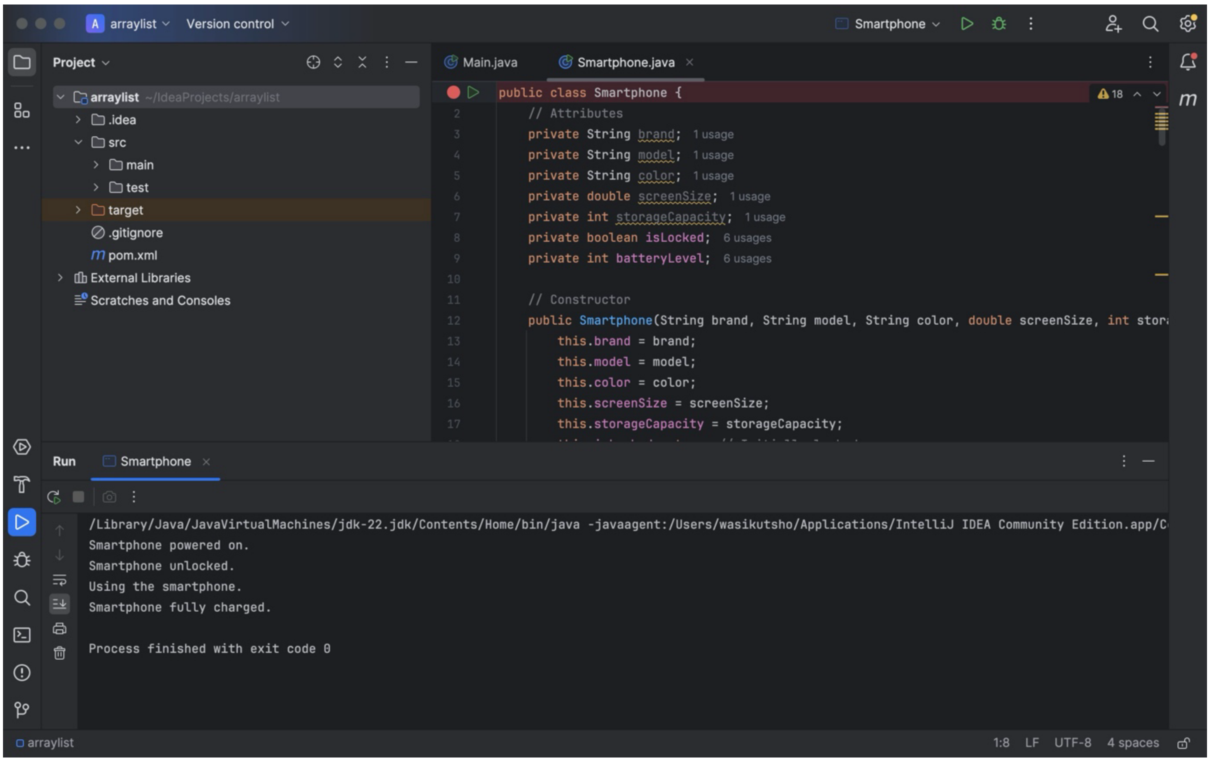
3/ Ans:



```java
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, absen
    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 bank
                    String taxID, int leaveBalance, String department) {
        this.employeeID = employeeID;
```

```
/Library/Java/JavaVirtualMachines/jdk-22.jdk/Contents/Home/bin/java -javaagent:/Users/wasikutsho/Applications/IntelliJ IDEA Community Edition.app/C
Monthly Salary: $5000.0
Salary deposited into bank account: 123456789
Pay Stub for John Doe:


Process finished with exit code 0
```

4/Ans:



```java
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 stora
        this.brand = brand;
        this.model = model;
        this.color = color;
        this.screenSize = screenSize;
        this.storageCapacity = storageCapacity;
```

```
/Library/Java/JavaVirtualMachines/jdk-22.jdk/Contents/Home/bin/java -javaagent:/Users/wasikutsho/Applications/IntelliJ IDEA Community Edition.app/C
Smartphone powered on.
Smartphone unlocked.
Using the smartphone.
Smartphone fully charged.

Process finished with exit code 0
```