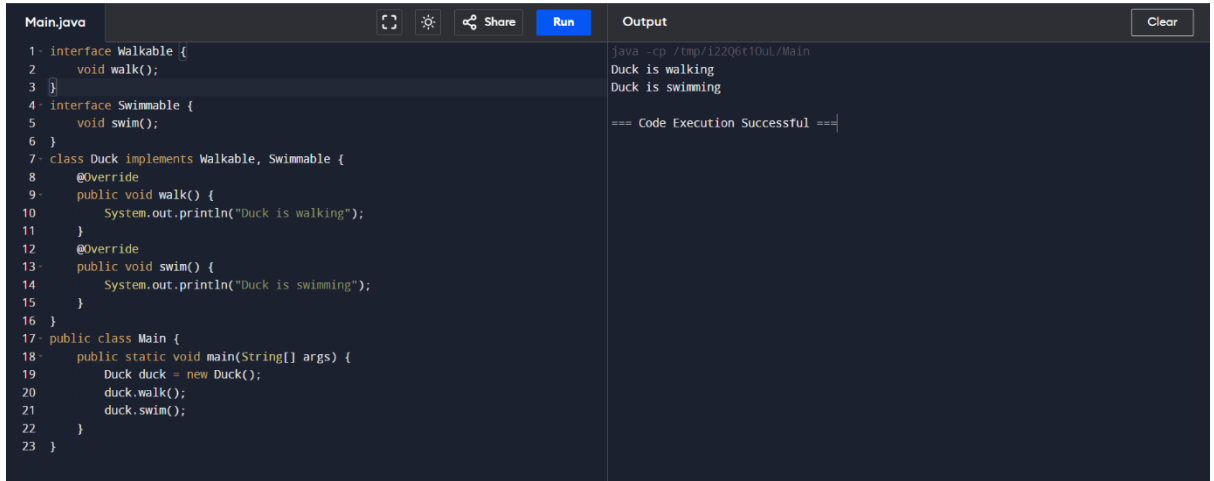


## 27/7/24 CLASS WORK

1. write a java program animal interface that takes no arguments and return void create dog class that implement animal and override speed method to print "dog in barking";

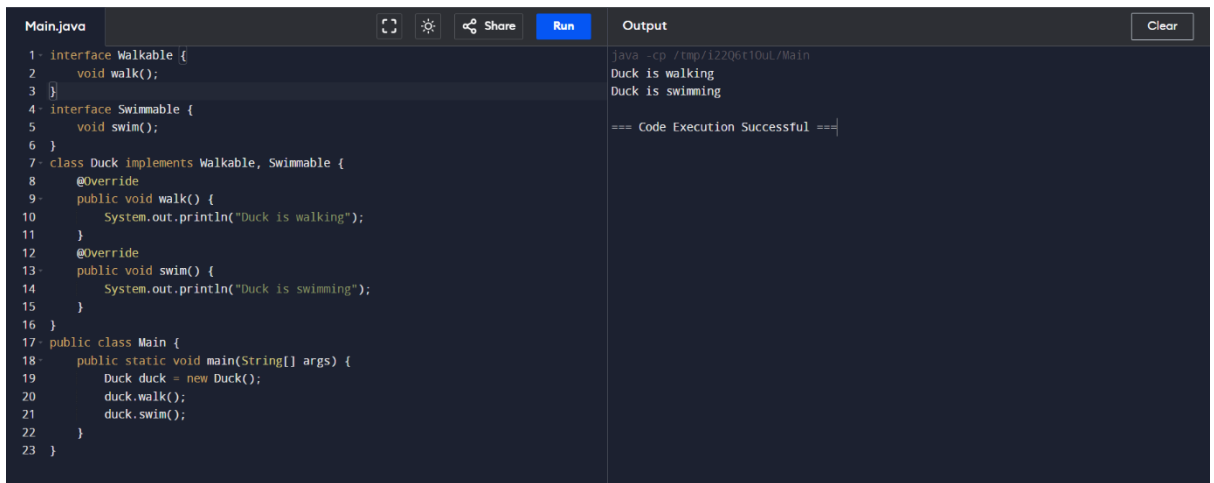


```
Main.java
1- interface Walkable {
2-     void walk();
3- }
4- interface Swimmable {
5-     void swim();
6- }
7- class Duck implements Walkable, Swimmable {
8-     @Override
9-     public void walk() {
10-         System.out.println("Duck is walking");
11-     }
12-     @Override
13-     public void swim() {
14-         System.out.println("Duck is swimming");
15-     }
16- }
17- public class Main {
18-     public static void main(String[] args) {
19-         Duck duck = new Duck();
20-         duck.walk();
21-         duck.swim();
22-     }
23- }
```

```
Output
java -cp /tmp/12206t10uL/Main
Duck is walking
Duck is swimming

=== Code Execution Successful ===
```

2. create interface walkable with walk method and create another interface with swimmable with swim method create class duck and implement the interface



```
Main.java
1- interface Walkable {
2-     void walk();
3- }
4- interface Swimmable {
5-     void swim();
6- }
7- class Duck implements Walkable, Swimmable {
8-     @Override
9-     public void walk() {
10-         System.out.println("Duck is walking");
11-     }
12-     @Override
13-     public void swim() {
14-         System.out.println("Duck is swimming");
15-     }
16- }
17- public class Main {
18-     public static void main(String[] args) {
19-         Duck duck = new Duck();
20-         duck.walk();
21-         duck.swim();
22-     }
23- }
```

```
Output
java -cp /tmp/12206t10uL/Main
Duck is walking
Duck is swimming

=== Code Execution Successful ===
```

3. create a java program to abstract class animal to abstract method sound create subclass lion and tiger that extend the animal class implement the sound method to make a specific sound for each animal

Main.java	Output
<pre>1- abstract class Animal { 2-     abstract void sound(); 3- } 4- class Lion extends Animal { 5-     @Override 6-     void sound() { 7-         System.out.println("Roarrrrr"); 8-     } 9- } 10- class Tiger extends Animal { 11-     @Override 12-     void sound(){ 13-         System.out.println("Growwwlllll"); 14-     } 15- } 16- public class Main{ 17-     public static void main(String[] args){ 18-         Animal lion=new Lion(); 19-         Animal tiger=new Tiger(); 20-         lion.sound(); 21-         tiger.sound(); 22-     } 23- } 24</pre>	<pre>java -cp ./tmp/DN32CsbMFZ/Main Roarrrrr Growwwlllll  === Code Execution Successful ===</pre>