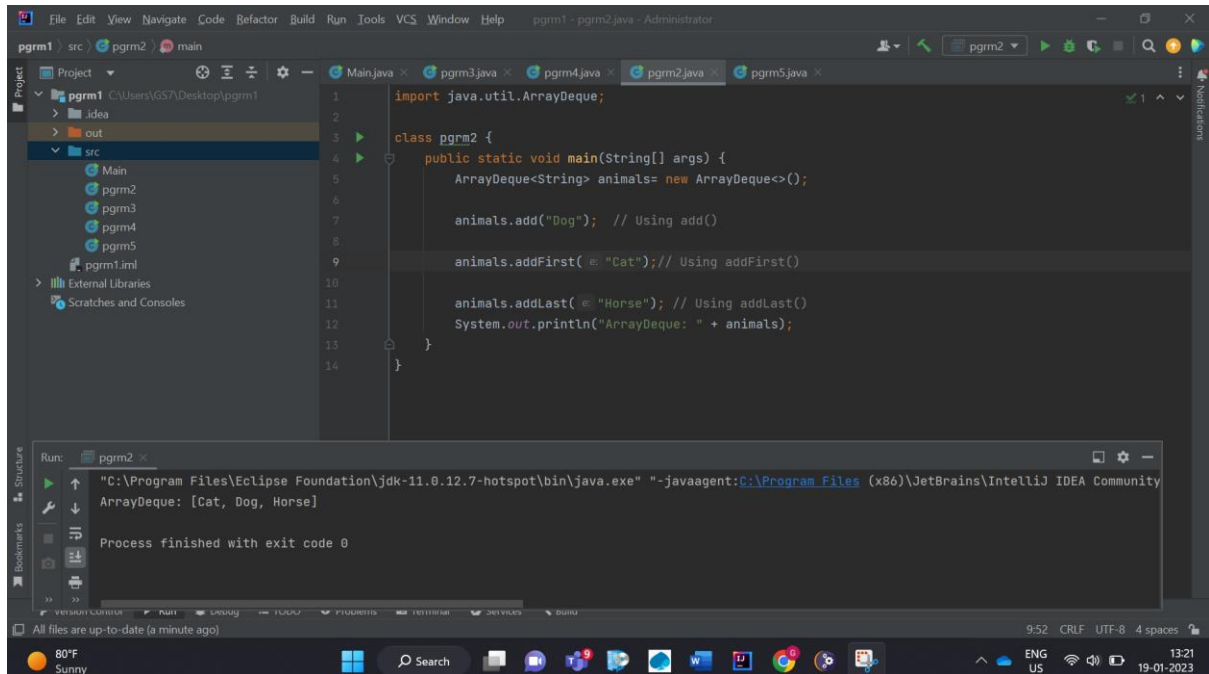


ArrayDeque Coding Questions

Task3: Code Snippets

1) Write a program to Insert Elements to Deque by using add(), addFirst() and addLast() methods.



The screenshot shows an IDE with a project named 'pgm1'. The source code for 'pgm2.java' is displayed, showing the insertion of 'Dog', 'Cat', and 'Horse' into an 'ArrayDeque'. The run console shows the output: 'ArrayDeque: [Cat, Dog, Horse]'.

```
import java.util.ArrayDeque;

class pgm2 {
    public static void main(String[] args) {
        ArrayDeque<String> animals= new ArrayDeque<>();

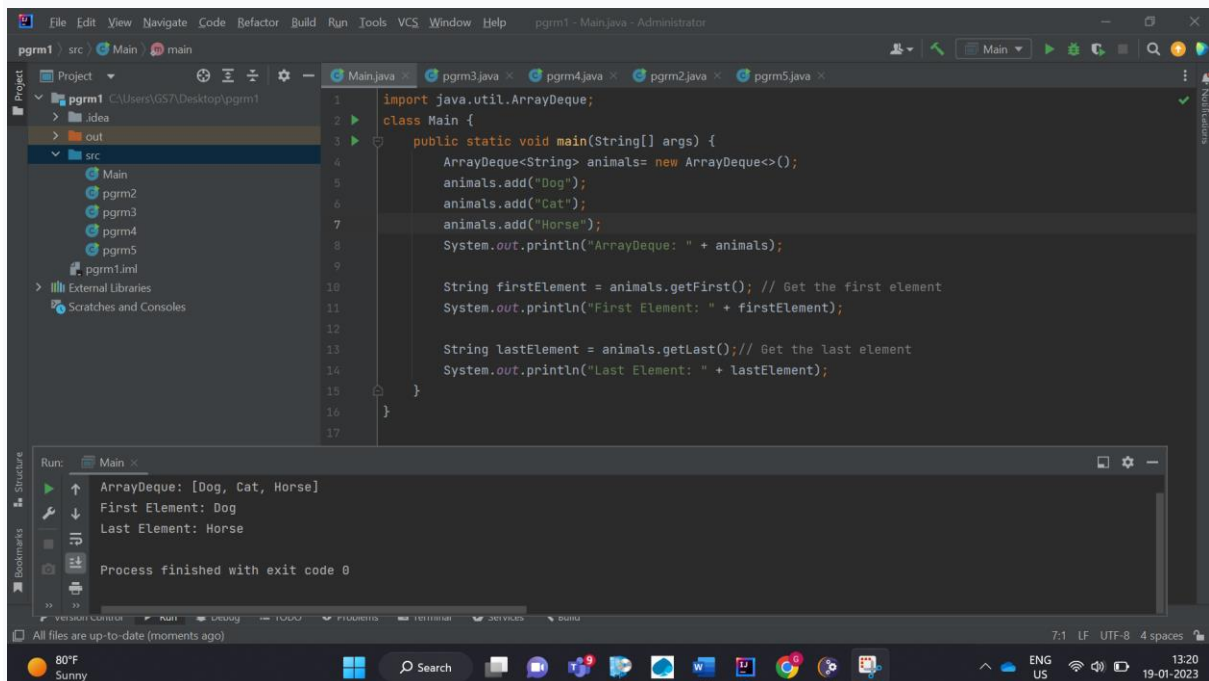
        animals.add("Dog"); // Using add()

        animals.addFirst("Cat");// Using addFirst()

        animals.addLast("Horse"); // Using addLast()
        System.out.println("ArrayDeque: " + animals);
    }
}
```

Run: pgm2 x
"C:\Program Files\Eclipse Foundation\jdk-11.0.12-hotspot\bin\java.exe" "-javaagent:C:\Program Files (x86)\JetBrains\IntelliJ IDEA Community
ArrayDeque: [Cat, Dog, Horse]
Process finished with exit code 0

2) Write a program to Access ArrayDeque Elements using getFirst() and getLast() methods.



The screenshot shows an IDE with a project named 'pgm1'. The source code for 'Main.java' is displayed, showing the insertion of 'Dog', 'Cat', and 'Horse' into an 'ArrayDeque', followed by retrieving the first and last elements. The run console shows the output: 'ArrayDeque: [Dog, Cat, Horse]', 'First Element: Dog', and 'Last Element: Horse'.

```
import java.util.ArrayDeque;

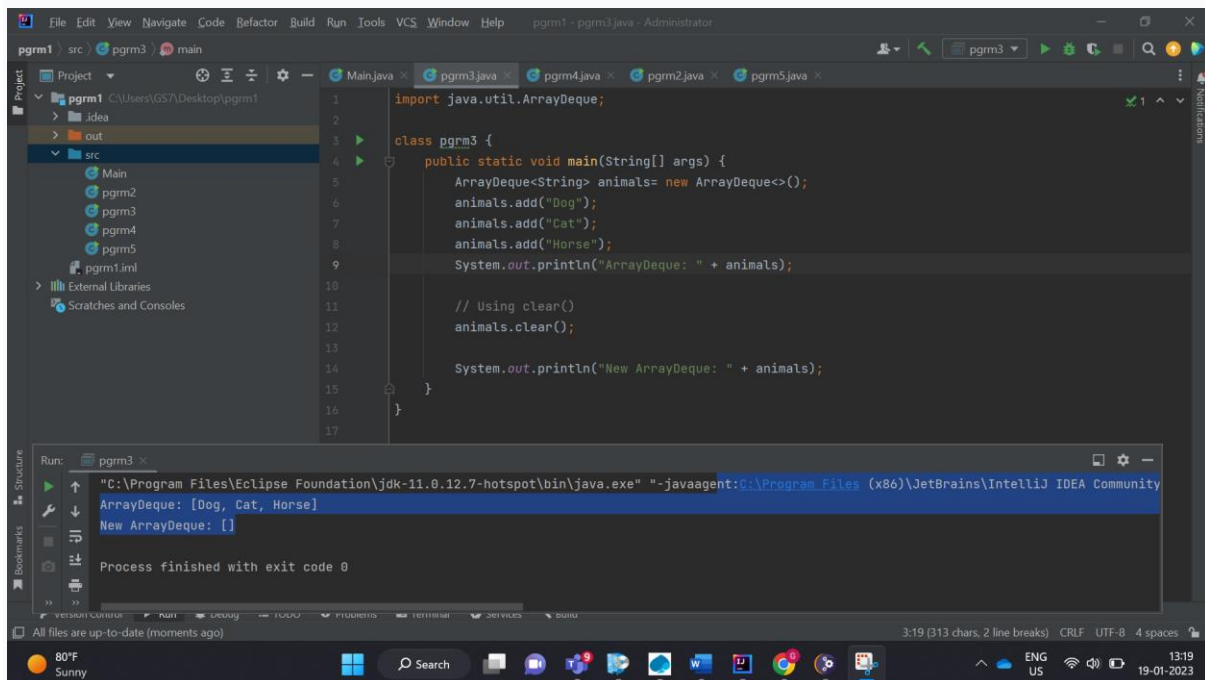
class Main {
    public static void main(String[] args) {
        ArrayDeque<String> animals= new ArrayDeque<>();
        animals.add("Dog");
        animals.add("Cat");
        animals.add("Horse");
        System.out.println("ArrayDeque: " + animals);

        String firstElement = animals.getFirst(); // Get the first element
        System.out.println("First Element: " + firstElement);

        String lastElement = animals.getLast(); // Get the last element
        System.out.println("Last Element: " + lastElement);
    }
}
```

Run: Main x
ArrayDeque: [Dog, Cat, Horse]
First Element: Dog
Last Element: Horse
Process finished with exit code 0

3) write a program to remove all the elements from the array deque using the clear() method.



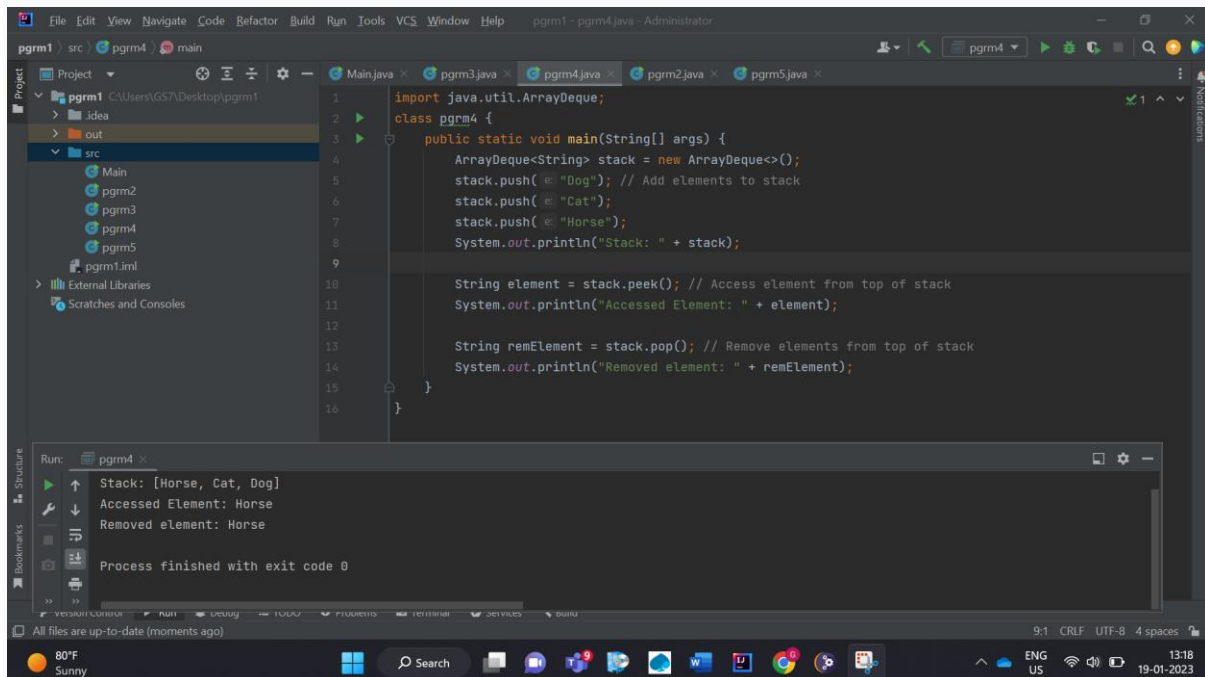
The screenshot shows the Eclipse IDE with a project named 'pgrm1'. The 'src' folder contains a file 'pgrm3.java'. The code in 'pgrm3.java' is as follows:

```
1 import java.util.ArrayDeque;
2
3 class pgrm3 {
4     public static void main(String[] args) {
5         ArrayDeque<String> animals= new ArrayDeque<>();
6         animals.add("Dog");
7         animals.add("Cat");
8         animals.add("Horse");
9         System.out.println("ArrayDeque: " + animals);
10
11         // Using clear()
12         animals.clear();
13
14         System.out.println("New ArrayDeque: " + animals);
15     }
16 }
17
```

The Run console shows the output of the program:

```
Run: pgrm3
"C:\Program Files\Eclipse Foundation\jdk-11.0.12.7-hotspot\bin\java.exe" "-javaagent:C:\Program Files (x86)\JetBrains\IntelliJ IDEA Community
ArrayDeque: [Dog, Cat, Horse]
New ArrayDeque: []
Process finished with exit code 0
```

4) Write a program to implement a **LIFO (Last-In-First-Out)** stacks in Java write a program to use a deque over the [Stack class](#) and show push(),pop(),peek() methods.



The screenshot shows the Eclipse IDE with a project named 'pgrm1'. The 'src' folder contains a file 'pgrm4.java'. The code in 'pgrm4.java' is as follows:

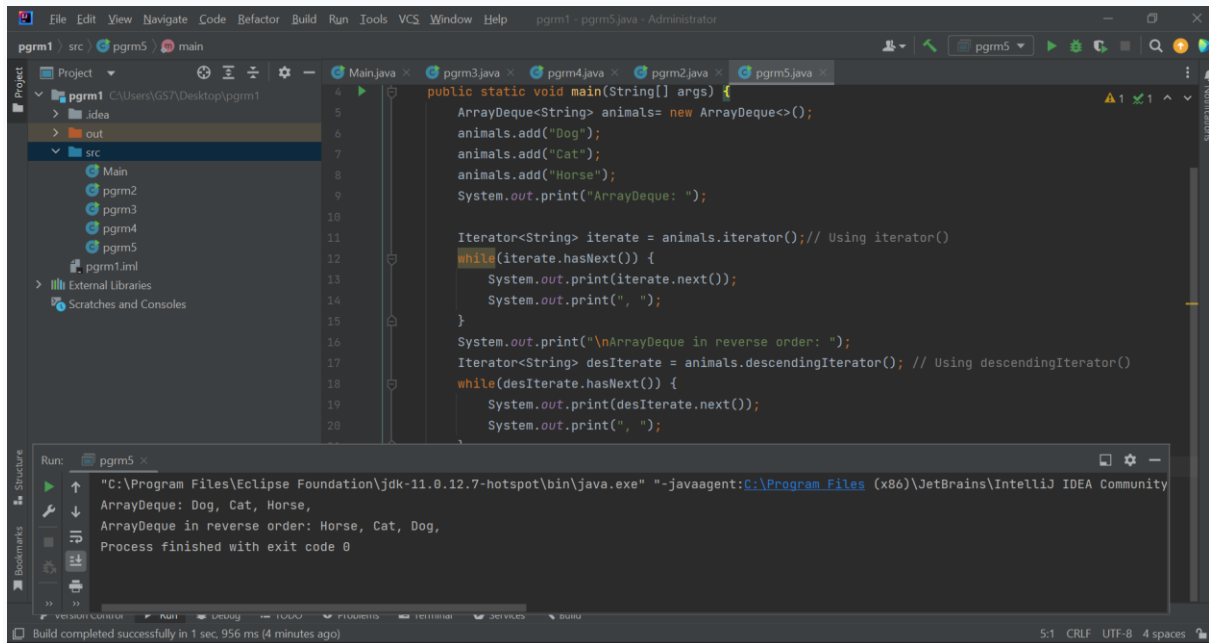
```
1 import java.util.ArrayDeque;
2
3 class pgrm4 {
4     public static void main(String[] args) {
5         ArrayDeque<String> stack = new ArrayDeque<>();
6         stack.push("Dog"); // Add elements to stack
7         stack.push("Cat");
8         stack.push("Horse");
9         System.out.println("Stack: " + stack);
10
11         String element = stack.peek(); // Access element from top of stack
12         System.out.println("Accessed Element: " + element);
13
14         String remElement = stack.pop(); // Remove elements from top of stack
15         System.out.println("Removed element: " + remElement);
16     }
17 }

```

The Run console shows the output of the program:

```
Run: pgrm4
Stack: [Horse, Cat, Dog]
Accessed Element: Horse
Removed element: Horse
Process finished with exit code 0
```

5) Write a program to implement iterating the ArrayDeque?



The screenshot shows an IDE window with a Java file named `pgrm5.java`. The code defines an `ArrayDeque` of strings, adds "Dog", "Cat", and "Horse", and then iterates through it using both a standard `Iterator` and a `descendingIterator`. The output in the Run console shows the elements in forward order and then in reverse order.

```
public static void main(String[] args) {  
    ArrayDeque<String> animals= new ArrayDeque<>();  
    animals.add("Dog");  
    animals.add("Cat");  
    animals.add("Horse");  
    System.out.print("ArrayDeque: ");  
  
    Iterator<String> iterate = animals.iterator();// Using iterator()  
    while(iterate.hasNext()) {  
        System.out.print(iterate.next());  
        System.out.print(", ");  
    }  
    System.out.print("\nArrayDeque in reverse order: ");  
    Iterator<String> desIterate = animals.descendingIterator(); // Using descendingIterator()  
    while(desIterate.hasNext()) {  
        System.out.print(desIterate.next());  
        System.out.print(", ");  
    }  
}
```

Run: pgrm5 x

```
"C:\Program Files\Eclipse Foundation\jdk-11.0.12-hotspot\bin\java.exe" "-javaagent:C:\Program Files (x86)\JetBrains\IntelliJ IDEA Community  
ArrayDeque: Dog, Cat, Horse,  
ArrayDeque in reverse order: Horse, Cat, Dog,  
Process finished with exit code 0
```

Build completed successfully in 1 sec. 956 ms (4 minutes ago)