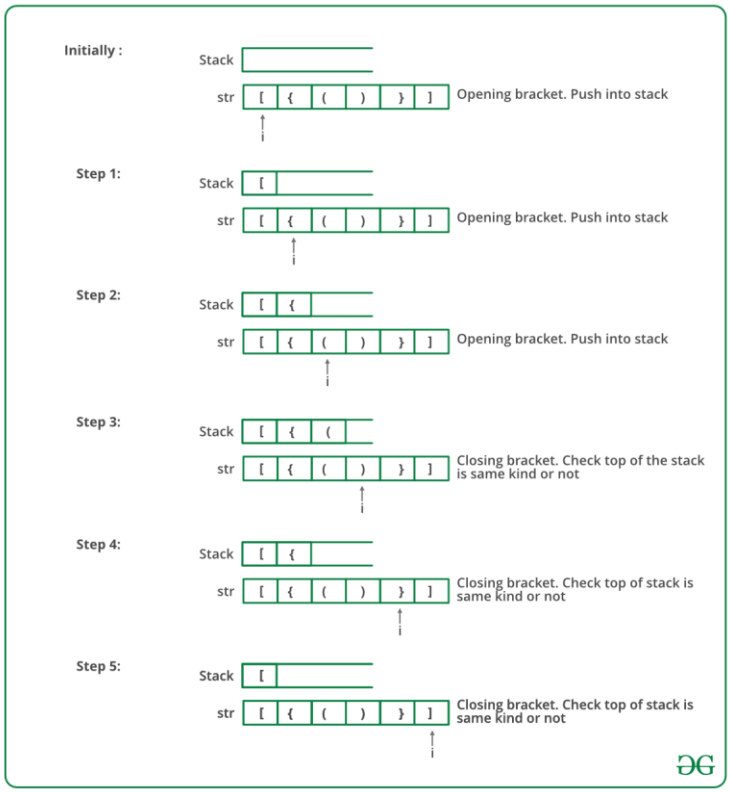


```
public class BracketProblem {
	public static void main(String[] args) {
		String str = "{[()}";
		if(checkBracket(str))
			System.out.println("The condition is valid");
		else
			System.out.println("In-valid Condition");
	}
	static boolean checkBracket(String str) {
		Deque<Character> stack = new ArrayDeque<>();
		for(int i=0;i<str.length();i++) {
			char temp = str.charAt(i);
			if(temp == '(' || temp =='{' || temp =='[')    {
				stack.push(temp);
				continue;                        }
```

```
                if(stack.isEmpty())
                        return false;
                char x;
                switch(temp) {
                case ')':
                        x=stack.pop();
                        if(x== '}' || x==']')
                                return false;
                        break;
                case ']':
                        x=stack.pop();
                        if(x== '}' || x==')')
                                return false;
                        break;
                case '}':
                        x=stack.pop();
                        if(x== ')' || x==']')
                                return false;
                        break;
                        }
                }
                return stack.isEmpty();
}
 }
```

ArrayDeque is part of the Java Collections Framework and is not written to be inherently thread safe.

Stack, together with Vector and Hashtable came with Java 1.0 and were implemented with thread safe operations (because it seemed like a good idea at the time). Acquiring and releasing thread locks is relatively expensive time wise, hence those data structures will be much slower