|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Experiment Title | Date | Sign |
| **1** | **Write a java program to implement stack and queue concept.** |  |  |
| **2** | **Write a java program to produce the tokens from given long string.** |  |  |
| **3** | **Write a java package to show dynamic polymorphism and interfaces.** |  |  |
| **4** | **Write a Java program to show multithreaded producer and consumer application.** |  |  |
| **5** | **Create a customized exception and also make use of all the 5 exception keywords.** |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**EXPERIMENT 1**

**Problem statement: Write a java program to implement stack and queue concept.**

**Programming Code:**

import java.util.\*;

class HelloWorld {

static void push\_stack(Stack<Integer> stack, int a) {

stack.push(a);

System.out.println("Pushed item " + a + " to the stack.");

}

static void pop\_stack(Stack<Integer> stack) {

while(!stack.isEmpty()) {

System.out.println("Popped item " + stack.pop() + " from the stack.");

}

}

static void push\_queue(Queue<Integer> queue, int a) {

queue.offer(a);

System.out.println("Pushed item " + a + " to the queue.");

}

static void pop\_queue(Queue<Integer> queue) {

while(!queue.isEmpty()) {

System.out.println("Popped item " + queue.poll() + " from the queue.");

}

}

public static void main(String[] args) {

System.out.println("----Stack----");

Stack<Integer> stack = new Stack<Integer>();

push\_stack(stack, 10);

push\_stack(stack, 20);

push\_stack(stack, 30);

pop\_stack(stack);

System.out.println("----Queue----");

Queue<Integer> queue = new LinkedList<>();

push\_queue(queue, 10);

push\_queue(queue, 20);

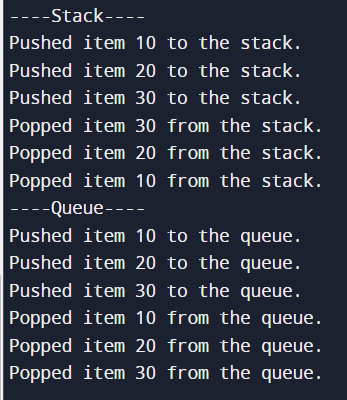
push\_queue(queue, 30);

pop\_queue(queue);

    }

}

**Output:**

****

**Learning Outcomes:**

**EXPERIMENT 2**

**Problem statement:** **Write a java program to produce the tokens from given long string.**

**Programming Code:**

import java.util.StringTokenizer;

public class Main {

public static void main (String [] args) {

String longString = "This is a long string with multiple words and punctuation.";

StringTokenizer tokenizer = new StringTokenizer(longString);

System.out.println("Tokens from the given long string:");

while (tokenizer.hasMoreTokens()) {

String token = tokenizer.nextToken();

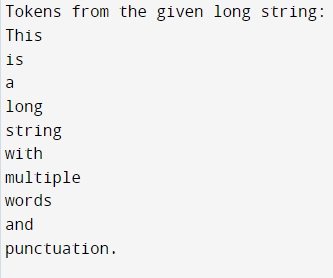
System.out.println(token);

}

}

}

**Output:**



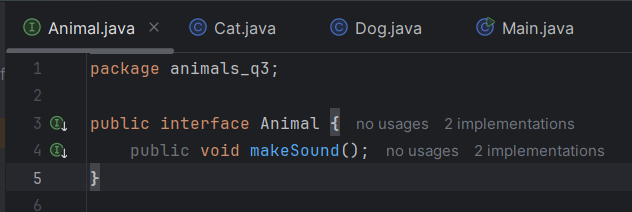
**Learning Outcomes:**

**EXPERIMENT 3**

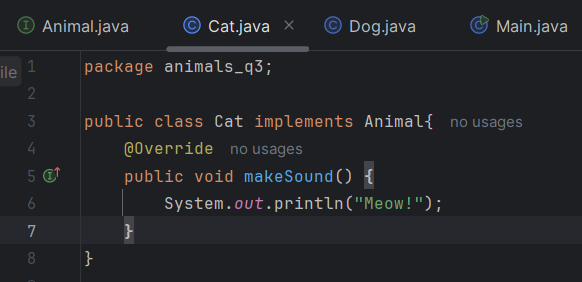
**Problem statement:** **Write a java package to show dynamic polymorphism and interfaces.**

**Programming Code:**

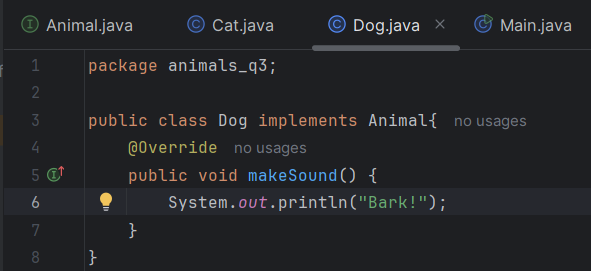
Interface(Animal.java)



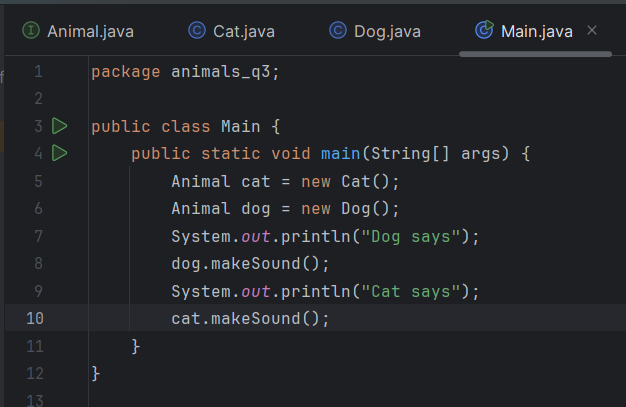
Cat.java



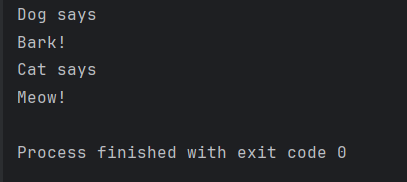
Dog.java



Main.java



**Output:**

****

**Learning Outcomes:**

**EXPERIMENT 4**

**Problem statement:** **Write a Java program to show multithreaded producer and consumer application.**

**Programming Code:**

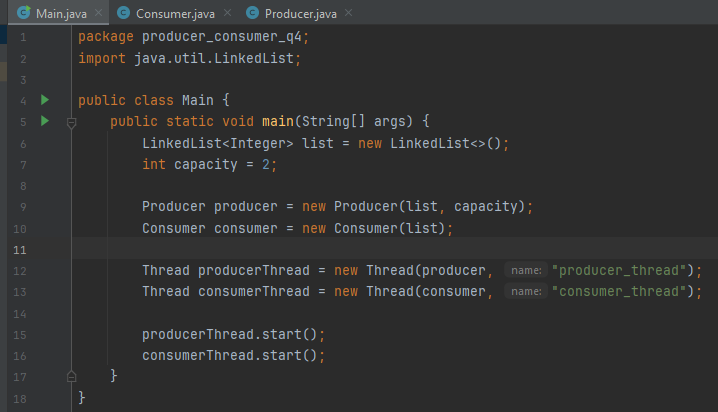
Producer.java



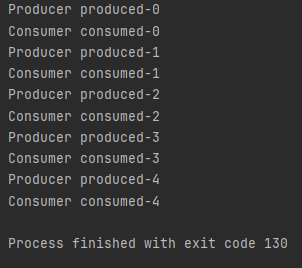
Consumer.java



Main.java



**Output:**

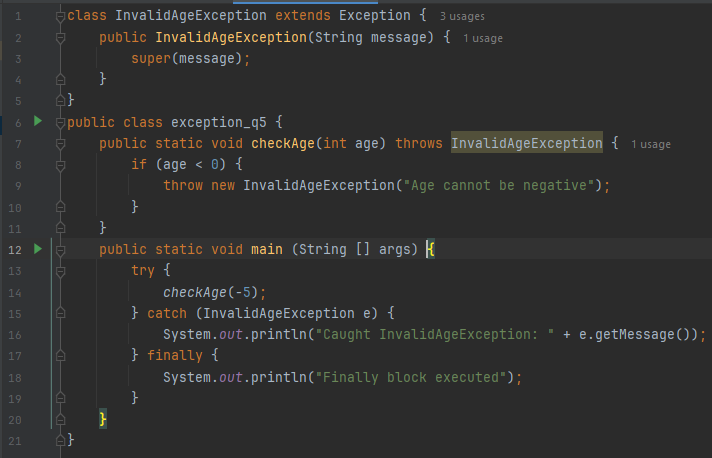
****

**Learning Outcomes:**

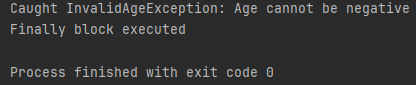
**EXPERIMENT 5**

**Problem statement:** **Create a customized exception and also make use of all the 5 exception keywords.**

**Programming Code:**



**Output:**

****

**Learning Outcomes:**