

PART-I (Pizza and Movies JUnit Tests)

```
1 package edu.cpp.cs5800.pizza;
2
3 import org.junit.jupiter.api.Test;
4
5 import java.util.ArrayList;
6 import java.util.List;
7
8 import static org.junit.jupiter.api.Assertions.assertEquals;
9
10 public class PizzaTest {
11
12     @Test
13     public void testPizzaSizeSmall() {
14         Pizza.Size size = Pizza.Size.SMALL;
15         assertEquals("Small", size.toString());
16     }
17
18     @Test
19     public void testPizzaSizeMedium() {
20         Pizza.Size size = Pizza.Size.MEDIUM;
21         assertEquals("Medium", size.toString());
22     }
23
24     @Test
25     public void testPizzaSizeLarge() {
26         Pizza.Size size = Pizza.Size.LARGE;
27         assertEquals("Large", size.toString());
28     }
29
30     @Test
31     public void testPizzaTopping() {
32         Pizza.Topping topping = Pizza.Topping.PEPPERONI;
33         assertEquals("Pepperoni", topping.toString());
34     }
35
36     @Test
37     public void testPizzaSize() {
38         List<Pizza.Topping> toppings = new ArrayList<>();
39         toppings.add(Pizza.Topping.EXTRA_CHEESE);
40         Pizza pizza = new Pizza(Pizza.Size.MEDIUM, toppings, "PizzaChain");
41         String expected = "Eating a Medium pizza at PizzaChain with 1 toppings: [Extra
42 Cheese]";
43         assertEquals(expected, pizza.eat());
44     }
45
46     @Test
47     public void testPizzaChainName() {
48         List<Pizza.Topping> toppings = new ArrayList<>();
49         toppings.add(Pizza.Topping.EXTRA_CHEESE);
50         Pizza pizza = new Pizza(Pizza.Size.MEDIUM, toppings, "ABC");
51         String expected = "Eating a Medium pizza at ABC with 1 toppings: [Extra Cheese]";
52         assertEquals(expected, pizza.eat());
53     }
54 }
```

```
53
54     @Test
55     public void testPizzaToppingCount() {
56         List<Pizza.Topping> toppings = new ArrayList<>();
57         toppings.add(Pizza.Topping.EXTRA_CHEESE);
58         toppings.add(Pizza.Topping.HAM);
59         Pizza pizza = new Pizza(Pizza.Size.MEDIUM, toppings, "ABC");
60         String expected = "Eating a Medium pizza at ABC with 2 toppings: [Extra Cheese,
Ham]";
61         assertEquals(expected, pizza.eat());
62     }
63
64     @Test
65     public void testPizzaToppings() {
66         List<Pizza.Topping> toppings = new ArrayList<>();
67         toppings.add(Pizza.Topping.EXTRA_CHEESE);
68         toppings.add(Pizza.Topping.HAM);
69         Pizza pizza = new Pizza(Pizza.Size.MEDIUM, toppings, "ABC");
70         String expected = "Eating a Medium pizza at ABC with 2 toppings: [Extra Cheese,
Ham]";
71         assertEquals(expected, pizza.eat());
72     }
73 }
74
```

PART-I (Pizza and Movies JUnit Tests)

```
1 package edu.cpp.cs5800.pizza;
2
3 import org.junit.jupiter.api.Test;
4
5 import static org.junit.jupiter.api.Assertions.assertEquals;
6
7 public class DominosTest {
8     @Test
9     public void testDominosName() {
10         Pizza pizza = (new Dominos()).setSize(Pizza.Size.SMALL).build();
11         String expected = "Eating a Small pizza at Dominos with 0 toppings: []";
12         assertEquals(expected, pizza.eat());
13     }
14 }
15
```

PART-I (Pizza and Movies JUnit Tests)

```
1 package edu.cpp.cs5800.pizza;
2
3 import org.junit.jupiter.api.Test;
4
5 import static org.junit.jupiter.api.Assertions.assertEquals;
6
7 public class PizzaHutTest {
8     @Test
9     public void testPizzaHutName() {
10         Pizza pizza = (new PizzaHut()).setSize(Pizza.Size.SMALL).build();
11         String expected = "Eating a Small pizza at Pizza Hut with 0 toppings: []";
12         assertEquals(expected, pizza.eat());
13     }
14 }
15
```

PART-I (Pizza and Movies JUnit Tests)

```
1 package edu.cpp.cs5800.pizza;
2
3 import org.junit.jupiter.api.Test;
4
5 import static org.junit.jupiter.api.Assertions.assertEquals;
6
7 class PizzaChainImpl extends PizzaChain {
8     @Override
9     public String getName() {
10         return "Pizza Chain";
11     }
12 }
13
14 public class PizzaChainTest {
15     private PizzaChain pizzaChain = new PizzaChainImpl();
16
17     @Test
18     public void testSetSize() {
19         Pizza pizza = pizzaChain.setSize(Pizza.Size.LARGE).build();
20         String expected = "Eating a Large pizza at Pizza Chain with 0 toppings: []";
21         assertEquals(expected, pizza.eat());
22     }
23
24     @Test
25     public void testAddPepperoni() {
26         Pizza pizza = pizzaChain.setSize(Pizza.Size.LARGE).addPepperoni().build();
27         String expected = "Eating a Large pizza at Pizza Chain with 1 toppings: [Pepperoni]";
28         ;
29         assertEquals(expected, pizza.eat());
30     }
31
32     @Test
33     public void testAddSausage() {
34         Pizza pizza = pizzaChain.setSize(Pizza.Size.LARGE).addSausage().build();
35         String expected = "Eating a Large pizza at Pizza Chain with 1 toppings: [Sausage]";
36         ;
37         assertEquals(expected, pizza.eat());
38     }
39
40     @Test
41     public void testAddMushrooms() {
42         Pizza pizza = pizzaChain.setSize(Pizza.Size.LARGE).addMushrooms().build();
43         String expected = "Eating a Large pizza at Pizza Chain with 1 toppings: [Mushrooms]";
44         ;
45         assertEquals(expected, pizza.eat());
46     }
47
48     @Test
49     public void testAddBacon() {
50         Pizza pizza = pizzaChain.setSize(Pizza.Size.LARGE).addBacon().build();
51         String expected = "Eating a Large pizza at Pizza Chain with 1 toppings: [Bacon]";
```

```

52     ;
53     assertEquals(expected, pizza.eat());
54 }
55
56 @Test
57 public void testAddOnions() {
58     Pizza pizza = pizzaChain.setSize(Pizza.Size.LARGE).addOnions().build();
59     String expected = "Eating a Large pizza at Pizza Chain with 1 toppings: [Onions]";
60     ;
61     assertEquals(expected, pizza.eat());
62 }
63
64 @Test
65 public void testExtraCheese() {
66     Pizza pizza = pizzaChain.setSize(Pizza.Size.LARGE).addExtraCheese().build();
67     String expected = "Eating a Large pizza at Pizza Chain with 1 toppings: [Extra
68 Cheese]";
69     ;
70     assertEquals(expected, pizza.eat());
71 }
72
73 @Test
74 public void testAddPeppers() {
75     Pizza pizza = pizzaChain.setSize(Pizza.Size.LARGE).addPeppers().build();
76     String expected = "Eating a Large pizza at Pizza Chain with 1 toppings: [Peppers]";
77     ;
78     assertEquals(expected, pizza.eat());
79 }
80
81 @Test
82 public void testAddChicken() {
83     Pizza pizza = pizzaChain.setSize(Pizza.Size.LARGE).addChicken().build();
84     String expected = "Eating a Large pizza at Pizza Chain with 1 toppings: [Chicken]";
85     ;
86     assertEquals(expected, pizza.eat());
87 }
88
89 @Test
90 public void testAddOlives() {
91     Pizza pizza = pizzaChain.setSize(Pizza.Size.LARGE).addOlives().build();
92     String expected = "Eating a Large pizza at Pizza Chain with 1 toppings: [Olives]";
93     ;
94     assertEquals(expected, pizza.eat());
95 }
96
97 @Test
98 public void testAddSpinach() {
99     Pizza pizza = pizzaChain.setSize(Pizza.Size.LARGE).addSpinach().build();
100    String expected = "Eating a Large pizza at Pizza Chain with 1 toppings: [Spinach]";
101    ;
102    ;

```

PART-I (Pizza and Movies JUnit Tests)

```

101     assertEquals(expected, pizza.eat());
102 }
103
104 @Test
105 public void testAddTomatoAndBasil() {
106     Pizza pizza = pizzaChain.setSize(Pizza.Size.LARGE).addTomatoAndBasil().build();
107     String expected = "Eating a Large pizza at Pizza Chain with 1 toppings: [Tomato
and Basil]";
108     ;
109     assertEquals(expected, pizza.eat());
110 }
111
112 @Test
113 public void testAddBeef() {
114     Pizza pizza = pizzaChain.setSize(Pizza.Size.LARGE).addBeef().build();
115     String expected = "Eating a Large pizza at Pizza Chain with 1 toppings: [Beef]";
116     ;
117     assertEquals(expected, pizza.eat());
118 }
119
120 @Test
121 public void testAddHam() {
122     Pizza pizza = pizzaChain.setSize(Pizza.Size.LARGE).addHam().build();
123     String expected = "Eating a Large pizza at Pizza Chain with 1 toppings: [Ham]";
124     ;
125     assertEquals(expected, pizza.eat());
126 }
127
128 @Test
129 public void testAddPesto() {
130     Pizza pizza = pizzaChain.setSize(Pizza.Size.LARGE).addPesto().build();
131     String expected = "Eating a Large pizza at Pizza Chain with 1 toppings: [Pesto]";
132     ;
133     assertEquals(expected, pizza.eat());
134 }
135
136 @Test
137 public void testAddSpicyPork() {
138     Pizza pizza = pizzaChain.setSize(Pizza.Size.LARGE).addSpicyPork().build();
139     String expected = "Eating a Large pizza at Pizza Chain with 1 toppings: [Spicy
Pork]";
140     ;
141     assertEquals(expected, pizza.eat());
142 }
143
144 @Test
145 public void testAddHamAndPineapple() {
146     Pizza pizza = pizzaChain.setSize(Pizza.Size.LARGE).addHamAndPineapple().build();
147     String
148         expected =
149         "Eating a Large pizza at Pizza Chain with 1 toppings: [Ham and Pineapple]"
150     ;
151     ;

```

PART-I (Pizza and Movies JUnit Tests)

```
151         assertEquals(expected, pizza.eat());
152     }
153
154     @Test
155     public void testAddingMultipleToppings() {
156         Pizza pizza = pizzaChain.setSize(Pizza.Size.LARGE)
157             .addBeef()
158             .addHamAndPineapple()
159             .addPesto()
160             .build();
161         String
162             expected =
163             "Eating a Large pizza at Pizza Chain with 3 toppings: [Beef, Ham and
164             Pineapple, Pesto]";
165         assertEquals(expected, pizza.eat());
166     }
167 }
168
```


PART-I (Pizza and Movies JUnit Tests)

```
1 package edu.cpp.cs5800.pizza;
2
3 import org.junit.platform.suite.api.SelectClasses;
4 import org.junit.platform.suite.api.Suite;
5
6 @Suite
7 @SelectClasses({
8     PizzaTestSuite.class,
9     DominosTest.class,
10    LittleCaesarsTest.class,
11    PizzaChainTest.class,
12    PizzaHutTest.class,
13    PizzaTest.class
14 })
15 public class PizzaTestSuite {
16 }
17
```

PART-I (Pizza and Movies JUnit Tests)

```
1 package edu.cpp.cs5800.pizza;
2
3 import org.junit.jupiter.api.Test;
4
5 import static org.junit.jupiter.api.Assertions.assertEquals;
6
7 public class LittleCaesarsTest {
8
9     @Test
10     public void testLittleCaesarsName() {
11         Pizza pizza = (new LittleCaesars()).setSize(Pizza.Size.SMALL).build();
12         String expected = "Eating a Small pizza at Little Caesars with 0 toppings: []";
13         assertEquals(expected, pizza.eat());
14     }
15 }
16
```