

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macroNutrient;
2
3 import edu.cpp.cs5800.macroNutrient.diet.Diet;
4 import edu.cpp.cs5800.macroNutrient.diet.Vegan;
5 import org.junit.jupiter.api.Test;
6
7 import static org.junit.jupiter.api.Assertions.assertEquals;
8
9 class FoodImpl extends Food {
10
11     @Override
12     public String getName() {
13         return "FoodName";
14     }
15
16     @Override
17     public String getType() {
18         return "FoodType";
19     }
20 }
21
22 public class FoodTest {
23     @Test
24     public void testToString() {
25         String expected = String.format("%s(%s)", "FoodType", "FoodName");
26         String actual = (new FoodImpl()).toString();
27         assertEquals(expected, actual);
28     }
29 }
30
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macronutrient;
2
3 import edu.cpp.cs5800.macronutrient.carbs.Bread;
4 import edu.cpp.cs5800.macronutrient.fats.Avocado;
5 import edu.cpp.cs5800.macronutrient.protein.Beef;
6 import org.junit.jupiter.api.Test;
7
8 import static org.junit.jupiter.api.Assertions.assertEquals;
9
10 public class MealTest {
11     @Test
12     public void testToString() {
13         String expected = "Meal: Carbs(Bread), Fats(Avocado), Protein(Beef)";
14         String actual = (new Meal(new Bread(), new Avocado(), new Beef())).toString();
15         assertEquals(expected, actual);
16     }
17
18     public static class NoRestrictionMealFactoryTest {
19     }
20 }
21
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macronutrient;
2
3 import edu.cpp.cs5800.macronutrient.diet.Diet;
4 import edu.cpp.cs5800.macronutrient.diet.Vegan;
5 import org.junit.jupiter.api.Test;
6
7 import static org.junit.jupiter.api.Assertions.assertEquals;
8
9 public class CustomerTest {
10     @Test
11     public void testToString() {
12         String customerName = "John Doe";
13         Diet diet = new Vegan();
14
15         String expected = "Customer (name=" + customerName + ", diet=" + diet + ")";
16         String actual = (new Customer(customerName, diet)).toString();
17         assertEquals(expected, actual);
18     }
19 }
20 }
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macronutrient;
2
3 import edu.cpp.cs5800.macronutrient.carbs.CarbsFactory;
4 import edu.cpp.cs5800.macronutrient.diet.NutAllergy;
5 import edu.cpp.cs5800.macronutrient.diet.Vegan;
6 import org.junit.jupiter.api.Test;
7
8 import java.util.List;
9
10 import static org.junit.jupiter.api.Assertions.assertEquals;
11
12 class FoodFactoryImpl extends FoodFactory {
13     private static FoodFactory instance;
14
15     @Override
16     public List<String> getItems() {
17         return List.of("Pistachio", "Peanuts", "Chicken");
18     }
19
20     public static FoodFactory getInstance() {
21         if (instance == null) {
22             instance = new FoodFactoryImpl();
23         }
24
25         return instance;
26     }
27 }
28
29 public class FoodFactoryTest {
30     @Test
31     public void testGetFoodName() {
32         FoodFactory foodFactory = FoodFactoryImpl.getInstance();
33         String expected = "Chicken";
34         String actual = foodFactory.getFoodName(new NutAllergy());
35         assertEquals(expected, actual);
36     }
37 }
38 }
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macroNutrient;
2
3 import edu.cpp.cs5800.macroNutrient.carbs.Carbs;
4 import edu.cpp.cs5800.macroNutrient.carbs.CarbsFactory;
5 import edu.cpp.cs5800.macroNutrient.diet.Paleo;
6 import edu.cpp.cs5800.macroNutrient.fats.Fats;
7 import edu.cpp.cs5800.macroNutrient.fats.FatsFactory;
8 import edu.cpp.cs5800.macroNutrient.protein.Protein;
9 import edu.cpp.cs5800.macroNutrient.protein.ProteinFactory;
10 import org.junit.jupiter.api.Test;
11
12 import java.util.List;
13
14 import static org.junit.jupiter.api.Assertions.assertFalse;
15 import static org.junit.jupiter.api.Assertions.assertTrue;
16
17 public class PaleoMealFactoryTest {
18     MacroNutrientFactory factory = PaleoMealFactory.getInstance();
19
20     @Test
21     public void TestCreateCarbs() {
22         Carbs carbs = factory.createCarbs();
23         List<String> allCarbs = CarbsFactory.getInstance().getItems();
24         List<String> nonEatables = (new Paleo()).getNonEatableItems();
25         assertTrue(allCarbs.contains(carbs.getName()));
26         assertFalse(nonEatables.contains(carbs.getName()));
27     }
28
29     @Test
30     public void TestCreateProtein() {
31         Protein protein = factory.createProtein();
32         List<String> allProtein = ProteinFactory.getInstance().getItems();
33         List<String> nonEatables = (new Paleo()).getNonEatableItems();
34         assertTrue(allProtein.contains(protein.getName()));
35         assertFalse(nonEatables.contains(protein.getName()));
36     }
37
38     @Test
39     public void TestCreateFats() {
40         Fats fats = factory.createFats();
41         List<String> allFats = FatsFactory.getInstance().getItems();
42         List<String> nonEatables = (new Paleo()).getNonEatableItems();
43         assertTrue(allFats.contains(fats.getName()));
44         assertFalse(nonEatables.contains(fats.getName()));
45     }
46 }
47 }
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macroNutrient;
2
3 import edu.cpp.cs5800.macroNutrient.carbs.Carbs;
4 import edu.cpp.cs5800.macroNutrient.carbs.CarbsFactory;
5 import edu.cpp.cs5800.macroNutrient.diet.Vegan;
6 import edu.cpp.cs5800.macroNutrient.fats.Fats;
7 import edu.cpp.cs5800.macroNutrient.fats.FatsFactory;
8 import edu.cpp.cs5800.macroNutrient.protein.Protein;
9 import edu.cpp.cs5800.macroNutrient.protein.ProteinFactory;
10 import org.junit.jupiter.api.Test;
11
12 import java.util.List;
13
14 import static org.junit.jupiter.api.Assertions.assertFalse;
15 import static org.junit.jupiter.api.Assertions.assertTrue;
16
17 public class VeganMealFactoryTest {
18     MacroNutrientFactory factory = VeganMealFactory.getInstance();
19
20     @Test
21     public void TestCreateCarbs() {
22         Carbs carbs = factory.createCarbs();
23         List<String> allCarbs = CarbsFactory.getInstance().getItems();
24         List<String> nonEatables = (new Vegan()).getNonEatableItems();
25         assertTrue(allCarbs.contains(carbs.getName()));
26         assertFalse(nonEatables.contains(carbs.getName()));
27     }
28
29     @Test
30     public void TestCreateProtein() {
31         Protein protein = factory.createProtein();
32         List<String> allProtein = ProteinFactory.getInstance().getItems();
33         List<String> nonEatables = (new Vegan()).getNonEatableItems();
34         assertTrue(allProtein.contains(protein.getName()));
35         assertFalse(nonEatables.contains(protein.getName()));
36     }
37
38     @Test
39     public void TestCreateFats() {
40         Fats fats = factory.createFats();
41         List<String> allFats = FatsFactory.getInstance().getItems();
42         List<String> nonEatables = (new Vegan()).getNonEatableItems();
43         assertTrue(allFats.contains(fats.getName()));
44         assertFalse(nonEatables.contains(fats.getName()));
45     }
46 }
47 }
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macroNutrient;
2
3 import edu.cpp.cs5800.macroNutrient.carbs.Carbs;
4 import edu.cpp.cs5800.macroNutrient.carbs.CarbsFactory;
5 import edu.cpp.cs5800.macroNutrient.diet.NutAllergy;
6 import edu.cpp.cs5800.macroNutrient.fats.Fats;
7 import edu.cpp.cs5800.macroNutrient.fats.FatsFactory;
8 import edu.cpp.cs5800.macroNutrient.protein.Protein;
9 import edu.cpp.cs5800.macroNutrient.protein.ProteinFactory;
10 import org.junit.jupiter.api.Test;
11
12 import java.util.List;
13
14 import static org.junit.jupiter.api.Assertions.assertFalse;
15 import static org.junit.jupiter.api.Assertions.assertTrue;
16
17 public class NutAllergyMealFactoryTest {
18     MacroNutrientFactory factory = NutAllergyMealFactory.getInstance();
19
20     @Test
21     public void TestCreateCarbs() {
22         Carbs carbs = factory.createCarbs();
23         List<String> allCarbs = CarbsFactory.getInstance().getItems();
24         List<String> nonEatables = (new NutAllergy()).getNonEatableItems();
25         assertTrue(allCarbs.contains(carbs.getName()));
26         assertFalse(nonEatables.contains(carbs.getName()));
27     }
28
29     @Test
30     public void TestCreateProtein() {
31         Protein protein = factory.createProtein();
32         List<String> allProtein = ProteinFactory.getInstance().getItems();
33         List<String> nonEatables = (new NutAllergy()).getNonEatableItems();
34         assertTrue(allProtein.contains(protein.getName()));
35         assertFalse(nonEatables.contains(protein.getName()));
36     }
37
38     @Test
39     public void TestCreateFats() {
40         Fats fats = factory.createFats();
41         List<String> allFats = FatsFactory.getInstance().getItems();
42         List<String> nonEatables = (new NutAllergy()).getNonEatableItems();
43         assertTrue(allFats.contains(fats.getName()));
44         assertFalse(nonEatables.contains(fats.getName()));
45     }
46 }
47 }
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macroNutrient;
2
3 import edu.cpp.cs5800.macroNutrient.carbs.Carbs;
4 import edu.cpp.cs5800.macroNutrient.carbs.CarbsFactory;
5 import edu.cpp.cs5800.macroNutrient.fats.Fats;
6 import edu.cpp.cs5800.macroNutrient.fats.FatsFactory;
7 import edu.cpp.cs5800.macroNutrient.protein.Protein;
8 import edu.cpp.cs5800.macroNutrient.protein.ProteinFactory;
9 import org.junit.jupiter.api.Test;
10
11 import java.util.List;
12
13 import static org.junit.jupiter.api.Assertions.assertTrue;
14
15 public class NoRestrictionMealFactoryTest {
16     MacroNutrientFactory factory = NoRestrictionMealFactory.getInstance();
17
18     @Test
19     public void TestCreateCarbs() {
20         Carbs carbs = factory.createCarbs();
21         List<String> allCarbs = CarbsFactory.getInstance().getItems();
22         assertTrue(allCarbs.contains(carbs.getName()));
23     }
24
25     @Test
26     public void TestCreateProtein() {
27         Protein protein = factory.createProtein();
28         List<String> allProtein = ProteinFactory.getInstance().getItems();
29         assertTrue(allProtein.contains(protein.getName()));
30     }
31
32     @Test
33     public void TestCreateFats() {
34         Fats fats = factory.createFats();
35         List<String> allFats = FatsFactory.getInstance().getItems();
36         assertTrue(allFats.contains(fats.getName()));
37     }
38 }
39
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macroNutrient;
2
3 import edu.cpp.cs5800.macroNutrient.diet.NoRestriction;
4 import edu.cpp.cs5800.macroNutrient.diet.Vegan;
5 import org.junit.jupiter.api.Test;
6
7 import static org.junit.jupiter.api.Assertions.assertInstanceOf;
8
9 public class MacroNutrientFactoryCreatorTest {
10
11     @Test
12     public void testCreateFactoryForNonDefault() {
13         MacroNutrientFactory factory = MacroNutrientFactoryCreator.createFactory(new Vegan
14         ());
15         assertInstanceOf(VeganMealFactory.class, factory);
16     }
17
18     @Test
19     public void testCreateFactoryForDefault() {
20         MacroNutrientFactory
21             factory =
22                 MacroNutrientFactoryCreator.createFactory(new NoRestriction());
23         assertInstanceOf(NoRestrictionMealFactory.class, factory);
24     }
25 }
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macronutrient.diet;
2
3 import org.junit.jupiter.api.Test;
4
5 import java.util.Arrays;
6 import java.util.List;
7
8 import static org.junit.jupiter.api.Assertions.assertEquals;
9
10 class DietImpl extends Diet {
11     @Override
12     public List<String> getNonEatableItems() {
13         return List.of("Abc", "Bcd");
14     }
15 }
16
17 public class DietTest {
18     @Test
19     public void testToString() {
20         String
21             expected =
22                 String.format("%s: Restricted%s", "DietImpl", Arrays.asList("Abc", "Bcd"));
23         String actual = (new DietImpl()).toString();
24         assertEquals(expected, actual);
25     }
26 }
27
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macronutrient.diet;
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 PaleoTest {
11     @Test
12     public void testGetNonEatableItems() {
13         List<String>
14             expected =
15             List.of("Cheese", "Bread", "Lentils", "Tofu", "Cheese", "Sour cream");
16         List<String> actual = (new Paleo()).getNonEatableItems();
17         assertEquals(expected, actual);
18     }
19 }
20
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macro.nutrient.diet;
2
3 import org.junit.jupiter.api.Test;
4
5 import java.util.List;
6
7 import static org.junit.jupiter.api.Assertions.assertEquals;
8
9 public class VeganTest {
10     @Test
11     public void testGetNonEatableItems() {
12         List<String> expected = List.of("Fish", "Chicken", "Tuna", "Cheese", "Sour cream");
13         List<String> actual = (new Vegan()).getNonEatableItems();
14         assertEquals(expected, actual);
15     }
16 }
17
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macro.nutrient.diet;
2
3 import org.junit.jupiter.api.Test;
4
5 import java.util.List;
6
7 import static org.junit.jupiter.api.Assertions.assertEquals;
8
9 public class NutAllergyTest {
10     @Test
11     public void testGetNonEatableItems() {
12         assertEquals(List.of("Pistachio", "Peanuts"), (new NutAllergy()).getNonEatableItems()
13             ());
14     }
15 }
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macro.nutrient.diet;
2
3 import org.junit.jupiter.api.Test;
4
5 import java.util.List;
6
7 import static org.junit.jupiter.api.Assertions.assertEquals;
8
9 public class NoRestrictionTest {
10     @Test
11     public void testGetNonEatableItems() {
12         assertEquals(List.of(), (new NoRestriction()).getNonEatableItems());
13     }
14 }
15
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macro.nutrient.fats;
2
3 import edu.cpp.cs5800.macro.nutrient.carbs.Carbs;
4 import org.junit.jupiter.api.Test;
5
6 import static org.junit.jupiter.api.Assertions.assertEquals;
7
8 class FatsImpl extends Fats {
9     @Override
10    public String getName() {
11        return "";
12    }
13 }
14
15 public class FatsTest {
16     @Test
17     public void testGetName() {
18         assertEquals("Fats", (new FatsImpl()).getType());
19     }
20 }
21
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macro.nutrient.fats;
2
3 import org.junit.jupiter.api.Test;
4
5 import static org.junit.jupiter.api.Assertions.assertEquals;
6
7 public class TunaTest {
8
9     @Test
10    public void testGetName() {
11        assertEquals("Tuna", (new Tuna()).getName());
12    }
13 }
14
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macro.nutrient.fats;
2
3 import org.junit.jupiter.api.Test;
4
5 import static org.junit.jupiter.api.Assertions.assertEquals;
6
7 public class AvocadoTest {
8
9     @Test
10    public void testGetName() {
11        assertEquals("Avocado", (new Avocado()).getName());
12    }
13 }
14
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macro.nutrient.fats;
2
3 import edu.cpp.cs5800.macro.nutrient.carbs.Lentils;
4 import org.junit.jupiter.api.Test;
5
6 import static org.junit.jupiter.api.Assertions.assertEquals;
7
8 public class LentilsTest {
9
10    @Test
11    public void testGetName() {
12        assertEquals("Lentils", (new Lentils()).getName());
13    }
14 }
15
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macro.nutrient.fats;
2
3 import org.junit.jupiter.api.Test;
4
5 import static org.junit.jupiter.api.Assertions.assertEquals;
6
7 public class PeanutsTest {
8
9     @Test
10    public void testGetName() {
11        assertEquals("Peanuts", (new Peanuts()).getName());
12    }
13 }
14
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macro.nutrient.fats;
2
3 import org.junit.jupiter.api.Test;
4
5 import static org.junit.jupiter.api.Assertions.assertEquals;
6
7 public class SourCreamTest {
8
9     @Test
10    public void testGetName() {
11        assertEquals("Sour cream", (new SourCream()).getName());
12    }
13 }
14
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macro.nutrient.fats;
2
3 import edu.cpp.cs5800.macro.nutrient.diet.Vegan;
4 import org.junit.jupiter.api.Test;
5
6 import java.util.List;
7
8 import static org.junit.jupiter.api.Assertions.assertEquals;
9 import static org.junit.jupiter.api.Assertions.assertTrue;
10
11 public class FatsFactoryTest {
12
13     private List<String> allFats = List.of("Avocado", "Peanuts", "Sour cream", "Tuna");
14     private FatsFactory factory = FatsFactory.getInstance();
15
16     @Test
17     public void TestGetFats() {
18         Fats fats = factory.getFats(new Vegan());
19         assertTrue(allFats.contains(fats.getName()));
20         assertEquals("Fats", fats.getType());
21     }
22
23     @Test
24     public void TestGetItems() {
25         List<String> items = factory.getItems();
26         assertEquals(allFats, items);
27     }
28 }
29
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macro.nutrient.carbs;
2
3 import org.junit.jupiter.api.Test;
4
5 import static org.junit.jupiter.api.Assertions.assertEquals;
6
7 public class BreadTest {
8
9     @Test
10    public void testGetName() {
11        assertEquals("Bread", (new Bread()).getName());
12    }
13 }
14
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macro.nutrient.carbs;
2
3 import org.junit.jupiter.api.Test;
4
5 import static org.junit.jupiter.api.Assertions.assertEquals;
6
7 class CarbsImpl extends Carbs {
8     @Override
9     public String getName() {
10         return "";
11     }
12 }
13
14 public class CarbsTest {
15     @Test
16     public void testGetName() {
17         assertEquals("Carbs", (new CarbsImpl()).getType());
18     }
19 }
20
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macro.nutrient.carbs;
2
3 import org.junit.jupiter.api.Test;
4
5 import static org.junit.jupiter.api.Assertions.assertEquals;
6
7 public class CheeseTest {
8
9     @Test
10    public void testGetName() {
11        assertEquals("Cheese", (new Cheese()).getName());
12    }
13 }
14
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macro.nutrient.carbs;
2
3 import org.junit.jupiter.api.Test;
4
5 import static org.junit.jupiter.api.Assertions.assertEquals;
6
7 public class LentilsTest {
8
9     @Test
10    public void testGetName() {
11        assertEquals("Lentils", (new Lentils()).getName());
12    }
13 }
14
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macro.nutrient.carbs;
2
3 import org.junit.jupiter.api.Test;
4
5 import static org.junit.jupiter.api.Assertions.assertEquals;
6
7 public class PistachioTest {
8
9     @Test
10    public void testGetName() {
11        assertEquals("Pistachio", (new Pistachio()).getName());
12    }
13 }
14
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macro.nutrient.carbs;
2
3 import edu.cpp.cs5800.macro.nutrient.diet.Vegan;
4 import org.junit.jupiter.api.Test;
5
6 import java.util.List;
7
8 import static org.junit.jupiter.api.Assertions.assertEquals;
9 import static org.junit.jupiter.api.Assertions.assertTrue;
10
11 public class CarbsFactoryTest {
12
13     private List<String> allCurbs = List.of("Cheese", "Bread", "Lentils", "Pistachio");
14     private CarbsFactory factory = CarbsFactory.getInstance();
15
16     @Test
17     public void TestGetCarbs() {
18         Carbs carbs = factory.getCarbs(new Vegan());
19         assertTrue(allCurbs.contains(carbs.getName()));
20         assertEquals("Carbs", carbs.getType());
21     }
22
23     @Test
24     public void TestGetItems() {
25         List<String> items = factory.getItems();
26         assertEquals(allCurbs, items);
27     }
28 }
29
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macro.nutrient.protein;
2
3 import org.junit.jupiter.api.Test;
4
5 import static org.junit.jupiter.api.Assertions.assertEquals;
6
7 public class BeefTest {
8
9     @Test
10    public void testGetName() {
11        assertEquals("Beef", (new Beef()).getName());
12    }
13 }
14
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macroprotein;
2
3 import org.junit.jupiter.api.Test;
4
5 import static org.junit.jupiter.api.Assertions.assertEquals;
6
7 public class FishTest {
8
9     @Test
10    public void testGetName() {
11        assertEquals("Fish", (new Fish()).getName());
12    }
13 }
14
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macroprotein.protein;
2
3 import org.junit.jupiter.api.Test;
4
5 import static org.junit.jupiter.api.Assertions.assertEquals;
6
7 public class TofuTest {
8
9     @Test
10    public void testGetName() {
11        assertEquals("Tofu", (new Tofu()).getName());
12    }
13 }
14
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macro.nutrient.protein;
2
3 import org.junit.jupiter.api.Test;
4
5 import static org.junit.jupiter.api.Assertions.assertEquals;
6
7 public class ChickenTest {
8
9     @Test
10    public void testGetName() {
11        assertEquals("Chicken", (new Chicken()).getName());
12    }
13 }
14
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macro.nutrient.protein;
2
3 import org.junit.jupiter.api.Test;
4
5 import static org.junit.jupiter.api.Assertions.assertEquals;
6
7 class ProteinImpl extends Protein {
8     @Override
9     public String getName() {
10         return "";
11     }
12 }
13
14 public class ProteinTest {
15     @Test
16     public void testGetName() {
17         assertEquals("Protein", (new ProteinImpl()).getType());
18     }
19 }
20
```

PART-2 (Macronutrient Meals JUnit)

```
1 package edu.cpp.cs5800.macroprotein.protein;
2
3 import edu.cpp.cs5800.macroprotein.diet.Vegan;
4 import org.junit.jupiter.api.Test;
5
6 import java.util.List;
7
8 import static org.junit.jupiter.api.Assertions.assertEquals;
9 import static org.junit.jupiter.api.Assertions.assertTrue;
10
11 public class ProteinFactoryTest {
12
13     private List<String> allProteins = List.of("Beef", "Chicken", "Fish", "Tofu");
14     private ProteinFactory factory = ProteinFactory.getInstance();
15
16     @Test
17     public void TestGetProtein() {
18         Protein protein = factory.getProtein(new Vegan());
19         assertTrue(allProteins.contains(protein.getName()));
20         assertEquals("Protein", protein.getType());
21     }
22
23     @Test
24     public void TestGetItems() {
25         List<String> items = factory.getItems();
26         assertEquals(allProteins, items);
27     }
28 }
29
```