

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macronutrient;
2
3 public abstract class Food {
4     public String toString() {
5         return String.format("%s(%s)", this.getType(), this.getName());
6     }
7
8     public abstract String getName();
9
10    public abstract String getType();
11 }
12
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macronutrient;
2
3 import edu.cpp.cs5800.macronutrient.carbs.Carbs;
4 import edu.cpp.cs5800.macronutrient.fats.Fats;
5 import edu.cpp.cs5800.macronutrient.protein.Protein;
6
7 public class Meal {
8     private Carbs carbs;
9     private Fats fats;
10    private Protein protein;
11
12    public Meal(Carbs carbs, Fats fats, Protein protein) {
13        this.carbs = carbs;
14        this.fats = fats;
15        this.protein = protein;
16    }
17
18    @Override
19    public String toString() {
20        return String.format("Meal: %s, %s, %s", this.carbs, this.fats, this.protein);
21    }
22}
23
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macronutrient;
2
3 import edu.cpp.cs5800.macronutrient.diet.Diet;
4
5 public class Customer {
6     private String name;
7     private Diet diet;
8
9     public Customer(String name, Diet diet) {
10         this.name = name;
11         this.diet = diet;
12     }
13
14     public String getName() {
15         return this.name;
16     }
17
18     public Diet getDiet() {
19         return this.diet;
20     }
21
22     @Override
23     public String toString() {
24         return "Customer (name=" + this.name + ", diet=" + this.diet + ")";
25     }
26 }
27 }
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macronutrient;
2
3 import edu.cpp.cs5800.macronutrient.diet.Diet;
4
5 import java.util.List;
6 import java.util.Random;
7
8 public abstract class FoodFactory {
9     private final Random rand = new Random();
10
11    public String getFoodName(Diet diet) {
12        List<String> nonEatableItems = diet.getNonEatableItems();
13        List<String> eatableItems = this.getItems()
14            .stream()
15            .filter(item -> !nonEatableItems.contains(item))
16            .toList();
17        return eatableItems.get(rand.nextInt(eatableItems.size()));
18    }
19
20    public abstract List<String> getItems();
21 }
22
```

PART-2 (Macronutrient Meals Implementation)

```
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.Diet;
6 import edu.cpp.cs5800.macronutrient.diet.NutAllergy;
7 import edu.cpp.cs5800.macronutrient.diet.Paleo;
8 import edu.cpp.cs5800.macronutrient.diet.Vegan;
9 import edu.cpp.cs5800.macronutrient.fats.Fats;
10 import edu.cpp.cs5800.macronutrient.fats.FatsFactory;
11 import edu.cpp.cs5800.macronutrient.protein.Protein;
12 import edu.cpp.cs5800.macronutrient.protein.ProteinFactory;
13
14 public class PaleoMealFactory extends MacronutrientFactory {
15     private static PaleoMealFactory instance;
16     private final Diet diet = new Paleo();
17
18     private PaleoMealFactory() {
19     }
20
21     @Override
22     public Carbs createCarbs() {
23         return CarbsFactory.getInstance().getCarbs(this.diet);
24     }
25
26     @Override
27     public Protein createProtein() {
28         return ProteinFactory.getInstance().getProtein(this.diet);
29     }
30
31     @Override
32     public Fats createFats() {
33         return FatsFactory.getInstance().getFats(this.diet);
34     }
35
36     public static PaleoMealFactory getInstance() {
37         if (instance == null) {
38             instance = new PaleoMealFactory();
39         }
40         return instance;
41     }
42 }
43 }
```

PART-2 (Macronutrient Meals Implementation)

```
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.Diet;
6 import edu.cpp.cs5800.macronutrient.diet.Vegan;
7 import edu.cpp.cs5800.macronutrient.fats.Fats;
8 import edu.cpp.cs5800.macronutrient.fats.FatsFactory;
9 import edu.cpp.cs5800.macronutrient.protein.Protein;
10 import edu.cpp.cs5800.macronutrient.protein.ProteinFactory;
11
12 public class VeganMealFactory extends MacronutrientFactory {
13     private static VeganMealFactory instance;
14     private final Diet diet = new Vegan();
15
16     private VeganMealFactory() {
17     }
18
19     @Override
20     public Carbs createCarbs() {
21         return CarbsFactory.getInstance().getCarbs(this.diet);
22     }
23
24     @Override
25     public Protein createProtein() {
26         return ProteinFactory.getInstance().getProtein(this.diet);
27     }
28
29     @Override
30     public Fats createFats() {
31         return FatsFactory.getInstance().getFats(this.diet);
32     }
33
34     public static VeganMealFactory getInstance() {
35         if (instance == null) {
36             instance = new VeganMealFactory();
37         }
38         return instance;
39     }
40 }
41 }
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macronutrient;
2
3 import edu.cpp.cs5800.macronutrient.carbs.Carbs;
4 import edu.cpp.cs5800.macronutrient.diet.NoRestriction;
5 import edu.cpp.cs5800.macronutrient.diet.NutAllergy;
6 import edu.cpp.cs5800.macronutrient.diet.Paleo;
7 import edu.cpp.cs5800.macronutrient.diet.Vegan;
8 import edu.cpp.cs5800.macronutrient.fats.Fats;
9 import edu.cpp.cs5800.macronutrient.protein.Protein;
10
11 public class MacroNutrientDriver {
12     public static void main(String[] args) {
13         Customer[] customers = new Customer[6];
14         customers[0] = new Customer("John", new NoRestriction());
15         customers[1] = new Customer("Robertson", new NutAllergy());
16         customers[2] = new Customer("Alicia", new Paleo());
17         customers[3] = new Customer("Paxton", new Vegan());
18         customers[4] = new Customer("Jimmy", new NoRestriction());
19         customers[5] = new Customer("Jayvon", new Paleo());
20
21         for (Customer customer : customers) {
22             MacronutrientFactory factory = MacroNutrientFactoryCreator.createFactory(
23                 customer.getDiet());
24             Carbs carbs = factory.createCarbs();
25             Fats fats = factory.createFats();
26             Protein protein = factory.createProtein();
27             Meal meal = new Meal(carbs, fats, protein);
28             String message = String.format("%s, %s", customer, meal);
29             System.out.println(message);
30         }
31     }
32 }
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macronutrient;
2
3 import edu.cpp.cs5800.macronutrient.carbs.Carbs;
4 import edu.cpp.cs5800.macronutrient.fats.Fats;
5 import edu.cpp.cs5800.macronutrient.protein.Protein;
6
7 public abstract class MacronutrientFactory {
8     public abstract Carbs createCarbs();
9     public abstract Protein createProtein();
10    public abstract Fats createFats();
11 }
12
```

PART-2 (Macronutrient Meals Implementation)

```
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.Diet;
6 import edu.cpp.cs5800.macronutrient.diet.NutAllergy;
7 import edu.cpp.cs5800.macronutrient.fats.Fats;
8 import edu.cpp.cs5800.macronutrient.fats.FatsFactory;
9 import edu.cpp.cs5800.macronutrient.protein.Protein;
10 import edu.cpp.cs5800.macronutrient.protein.ProteinFactory;
11
12 public class NutAllergyMealFactory extends MacronutrientFactory {
13     private static NutAllergyMealFactory instance;
14     private final Diet diet = new NutAllergy();
15
16     private NutAllergyMealFactory() {
17     }
18
19     @Override
20     public Carbs createCarbs() {
21         return CarbsFactory.getInstance().getCarbs(this.diet);
22     }
23
24     @Override
25     public Protein createProtein() {
26         return ProteinFactory.getInstance().getProtein(this.diet);
27     }
28
29     @Override
30     public Fats createFats() {
31         return FatsFactory.getInstance().getFats(this.diet);
32     }
33
34     public static NutAllergyMealFactory getInstance() {
35         if (instance == null) {
36             instance = new NutAllergyMealFactory();
37         }
38         return instance;
39     }
40 }
41 }
```

PART-2 (Macronutrient Meals Implementation)

```
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.Diet;
6 import edu.cpp.cs5800.macronutrient.diet.NoRestriction;
7 import edu.cpp.cs5800.macronutrient.fats.Fats;
8 import edu.cpp.cs5800.macronutrient.fats.FatsFactory;
9 import edu.cpp.cs5800.macronutrient.protein.Protein;
10 import edu.cpp.cs5800.macronutrient.protein.ProteinFactory;
11
12 public class NoRestrictionMealFactory extends MacronutrientFactory {
13     private static NoRestrictionMealFactory instance;
14     private final Diet diet = new NoRestriction();
15
16     private NoRestrictionMealFactory() {
17     }
18
19     @Override
20     public Carbs createCarbs() {
21         return CarbsFactory.getInstance().getCarbs(this.diet);
22     }
23
24     @Override
25     public Protein createProtein() {
26         return ProteinFactory.getInstance().getProtein(this.diet);
27     }
28
29     @Override
30     public Fats createFats() {
31         return FatsFactory.getInstance().getFats(this.diet);
32     }
33
34     public static NoRestrictionMealFactory getInstance() {
35         if (instance == null) {
36             instance = new NoRestrictionMealFactory();
37         }
38         return instance;
39     }
40 }
41 }
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macronutrient;
2
3 import edu.cpp.cs5800.macronutrient.diet.Diet;
4 import edu.cpp.cs5800.macronutrient.diet.NutAllergy;
5 import edu.cpp.cs5800.macronutrient.diet.Paleo;
6 import edu.cpp.cs5800.macronutrient.diet.Vegan;
7
8 public class MacroNutrientFactoryCreator {
9     public static MacronutrientFactory createFactory(Diet diet) {
10         return switch (diet) {
11             case Vegan vegan -> VeganMealFactory.getInstance();
12             case NutAllergy nutAllergy -> NutAllergyMealFactory.getInstance();
13             case Paleo paleo -> PaleoMealFactory.getInstance();
14             case null, default -> NoRestrictionMealFactory.getInstance();
15         };
16     }
17 }
18 }
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macronutrient.diet;
2
3 import java.util.Arrays;
4 import java.util.List;
5
6 public abstract class Diet {
7     public abstract List<String> getNonEatableItems();
8     @Override
9     public String toString() {
10         return String.format("%s: Restricted%s", this.getClass().getSimpleName(), Arrays.
11             toString(this.getNonEatableItems().toArray()));
12     }
13 }
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macronutrient.diet;
2
3 import java.util.List;
4
5 public class Paleo extends Diet {
6     @Override
7     public List<String> getNonEatableItems() {
8         return List.of("Cheese", "Bread", "Lentils", "Tofu", "Cheese", "Sour cream");
9     }
10}
11
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macronutrient.diet;
2
3 import java.util.List;
4
5 public class Vegan extends Diet {
6     @Override
7     public List<String> getNonEatableItems() {
8         return List.of("Fish", "Chicken", "Tuna", "Cheese", "Sour cream");
9     }
10}
11
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macronutrient.diet;
2
3 import java.util.List;
4
5 public class NutAllergy extends Diet {
6     @Override
7     public List<String> getNonEatableItems() {
8         return List.of("Pistachio", "Peanuts");
9     }
10}
11
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macronutrient.diet;
2
3 import java.util.List;
4
5 public class NoRestriction extends Diet {
6     @Override
7     public List<String> getNonEatableItems() {
8         return List.of();
9     }
10 }
11
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macroNutrient.fats;  
2  
3 import edu.cpp.cs5800.macroNutrient.Food;  
4  
5 public abstract class Fats extends Food {  
6     private final static String type = "Fats";  
7  
8     public String getType() {  
9         return type;  
10    }  
11}  
12}  
13
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macronutrient.fats;  
2  
3 public class Tuna extends Fats {  
4     private final static String name = "Tuna";  
5  
6     public String getName() {  
7         return name;  
8     }  
9 }  
10
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macronutrient.fats;  
2  
3 public class Avocado extends Fats {  
4     private final static String name = "Avocado";  
5  
6     public String getName() {  
7         return name;  
8     }  
9 }  
10
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macronutrient.fats;  
2  
3 public class Peanuts extends Fats {  
4     private final static String name = "Peanuts";  
5  
6     public String getName() {  
7         return name;  
8     }  
9 }  
10
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macronutrient.fats;  
2  
3 public class SourCream extends Fats {  
4     private final static String name = "Sour cream";  
5  
6     public String getName() {  
7         return name;  
8     }  
9 }  
10
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macronutrient.fats;
2
3 import edu.cpp.cs5800.macronutrient.FoodFactory;
4 import edu.cpp.cs5800.macronutrient.diet.Diet;
5
6 import java.util.List;
7
8 public class FatsFactory extends FoodFactory {
9     private static FatsFactory instance;
10
11    private FatsFactory() {
12    }
13
14    public Fats getFats(Diet diet) {
15        String foodName = this.getFoodName(diet);
16        return createFats(foodName);
17    }
18
19    private Fats createFats(String carbsName) {
20        switch (carbsName) {
21            case "Avocado" -> {
22                return new Avocado();
23            }
24            case "Peanuts" -> {
25                return new Peanuts();
26            }
27            case "Sour cream" -> {
28                return new SourCream();
29            }
30            default -> {
31                return new Tuna();
32            }
33        }
34    }
35
36
37    public List<String> getItems() {
38        return List.of("Avocado", "Peanuts", "Sour cream", "Tuna");
39    }
40
41    public static FatsFactory getInstance() {
42        if (instance == null) {
43            instance = new FatsFactory();
44        }
45
46        return instance;
47    }
48 }
49 }
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macroNutrient.carbs;  
2  
3 public class Bread extends Carbs {  
4     private final static String name = "Bread";  
5  
6     public String getName() {  
7         return name;  
8     }  
9 }  
10
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macroNutrient.carbs;  
2  
3 import edu.cpp.cs5800.macroNutrient.Food;  
4  
5 public abstract class Carbs extends Food {  
6     private final static String type = "Carbs";  
7  
8     public String getType() {  
9         return type;  
10    }  
11 }  
12
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macronutrient.carbs;  
2  
3 public class Cheese extends Carbs {  
4     private final static String name = "Cheese";  
5  
6     public String getName() {  
7         return name;  
8     }  
9 }  
10
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macroNutrient.carbs;  
2  
3 public class Lentils extends Carbs {  
4     private final static String name = "Lentils";  
5  
6     public String getName() {  
7         return name;  
8     }  
9 }  
10
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macronutrient.carbs;  
2  
3 public class Pistachio extends Carbs {  
4     private final static String name = "Pistachio";  
5  
6     public String getName() {  
7         return name;  
8     }  
9 }  
10
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macroNutrient.carbs;
2
3 import edu.cpp.cs5800.macroNutrient.FoodFactory;
4 import edu.cpp.cs5800.macroNutrient.diet.Diet;
5
6 import java.util.List;
7
8 public class CarbsFactory extends FoodFactory {
9     private static CarbsFactory instance;
10
11    private CarbsFactory() {
12    }
13
14    public Carbs getCarbs(Diet diet) {
15        String foodName = this.getFoodName(diet);
16        return createCarbs(foodName);
17    }
18
19    private Carbs createCarbs(String carbsName) {
20        switch (carbsName) {
21            case "Cheese" -> {
22                return new Cheese();
23            }
24            case "Bread" -> {
25                return new Bread();
26            }
27            case "Lentils" -> {
28                return new Lentils();
29            }
30            default -> {
31                return new Pistachio();
32            }
33        }
34    }
35
36    public List<String> getItems() {
37        return List.of("Cheese", "Bread", "Lentils", "Pistachio");
38    }
39
40    public static CarbsFactory getInstance() {
41        if (instance == null) {
42            instance = new CarbsFactory();
43        }
44
45        return instance;
46    }
47 }
48 }
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macronutrient.protein;  
2  
3 public class Beef extends Protein {  
4     private final static String name = "Beef";  
5  
6     public String getName() {  
7         return name;  
8     }  
9 }  
10
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macronutrient.protein;  
2  
3 public class Fish extends Protein {  
4     private final static String name = "Fish";  
5  
6     public String getName() {  
7         return name;  
8     }  
9 }  
10
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macronutrient.protein;  
2  
3 public class Tofu extends Protein {  
4     private final static String name = "Tofu";  
5  
6     public String getName() {  
7         return name;  
8     }  
9 }  
10
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macronutrient.protein;  
2  
3 public class Chicken extends Protein {  
4     private final static String name = "Chicken";  
5  
6     public String getName() {  
7         return name;  
8     }  
9 }  
10
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macroNutrient.protein;  
2  
3 import edu.cpp.cs5800.macroNutrient.Food;  
4  
5 public abstract class Protein extends Food {  
6     private final static String type = "Protein";  
7  
8     public String getType() {  
9         return type;  
10    }  
11 }  
12
```

PART-2 (Macronutrient Meals Implementation)

```
1 package edu.cpp.cs5800.macroNutrient.protein;
2
3 import edu.cpp.cs5800.macroNutrient.FoodFactory;
4 import edu.cpp.cs5800.macroNutrient.diet.Diet;
5
6 import java.util.List;
7
8 public class ProteinFactory extends FoodFactory {
9     private static ProteinFactory instance;
10
11    private ProteinFactory() {
12    }
13
14    public Protein getProtein(Diet diet) {
15        String foodName = this.getFoodName(diet);
16        return createProtein(foodName);
17    }
18
19    private Protein createProtein(String carbsName) {
20        switch (carbsName) {
21            case "Beef" -> {
22                return new Beef();
23            }
24            case "Chicken" -> {
25                return new Chicken();
26            }
27            case "Fish" -> {
28                return new Fish();
29            }
30            default -> {
31                return new Tofu();
32            }
33        }
34    }
35
36    public List<String> getItems() {
37        return List.of("Beef", "Chicken", "Fish", "Tofu");
38    }
39
40    public static ProteinFactory getInstance() {
41        if (instance == null) {
42            instance = new ProteinFactory();
43        }
44
45        return instance;
46    }
47 }
48 }
```