|  |  |
| --- | --- |
| **Practicum Case** |  |
| COMP6122 | COMP6122001  Framework Layer Architecture |
| **Computer Science** | **O221-COMP6122-CT01-02** |
| ***Valid on*** *Odd Semester Year 2021/2022* | **Revision 00** |

**Learning Outcome**

* LO1 – describe use of design pattern in java
* LO2 – apply design pattern in java

**Topic**

* Session 2 – Creational Design Pattern I

## Sub Topics

* Singleton
* Factory Method

1. **Singleton**

Singleton is a creational design pattern that lets you ensure that a class has only one instance, while providing a global access point to this instance. If your code has access to the Singleton class, then it’s able to call the Singleton’s static method. So, whenever that method is called, the same object is always returned.

1. **Factory**

Factory Method is a creational design pattern that provides an interface for creating objects in a superclass but allows subclasses to alter the type of objects that will be created. The Factory Method pattern suggests that you replace direct object construction calls (using the new operator) with calls to a special factory method. Don’t worry: the objects are still created via the new operator, but it’s being called from within the factory method. Objects returned by a factory method are often referred to as products.

## Soal

*Case*

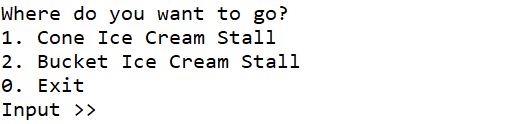
**Toppo Shop**

Toppo Shop is a popular ice cream shop in Jakarta. Every day the Toppo shop sells a lot of cone ice cream and rapidly grown business, the business also grown into two stalls in a shop that sells **cone ice cream** and another one that sells **bucket ice cream**. Because of that, maybe in the future will be some new stalls are coming in. Angel as the business owner of the shop, she asks you to create a program. Below are the following details of the program:

* **Main Menu**

At first, the program will show the menu. In this menu, the program will ask the user to choose **between** **0 and 2 (inclusive)**.

1. **Cone Ice Cream Stall**
2. **Bucket Ice Cream Stall**
3. **Exit**

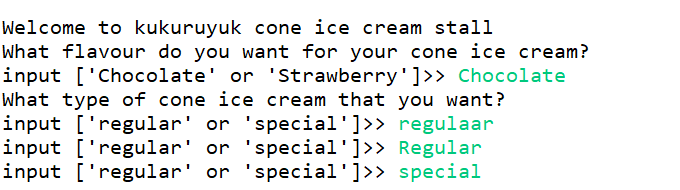


**Figure 1. Main Menu**

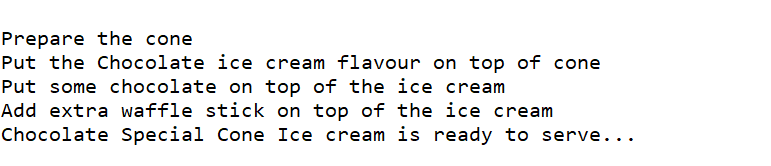
* 1. **Cone Ice Cream Stall**

If the customer **choose** **1**, then the program will:

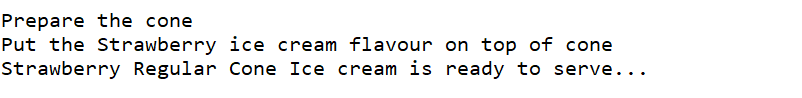
* Ask the user to input the **flavor** and validate the flavor must **either** “**Strawberry**” or “**Chocolate**”.
* Ask the user to input the **type of ice cream** that he/she wants and validate the type must **either** “**regular**”or “**special**”.
* After that, the program will **generate** the **cone ice cream** based on his/her flavor and the type. After that, **save** the generated ice cream to the **history of purchase** by user.



**Figure 2. Cone Ice Cream Input**



**Figure 3. Generate Special Cone Ice Cream**



**Figure 4. Generate Regular Cone Ice Cream**

* 1. **Bucket Ice Cream Stall**

If the customer **choose** **2**, then the program will:

* Ask the user to input the **flavor** and validate the flavor must **either** “**Strawberry**” or “**Chocolate**”.
* Ask the user to input the **type of ice cream** that he/she wants and validate the type must **either** “**regular**”or “**special**”.
* After that, the program will **generate** the **bucket ice cream** based on his/her flavor and the type. After that, **save** the generated ice cream to the **history of purchase** by user.

Text

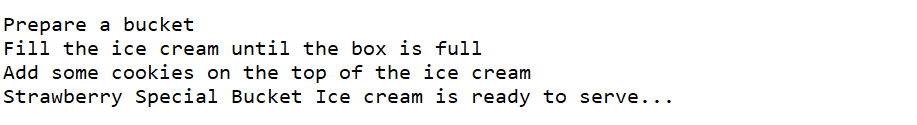
Description automatically generated

**Figure 5. Bucket Ice Cream Input**

Text

Description automatically generated with low confidence

**Figure 6. Generate Regular Bucket Ice Cream**

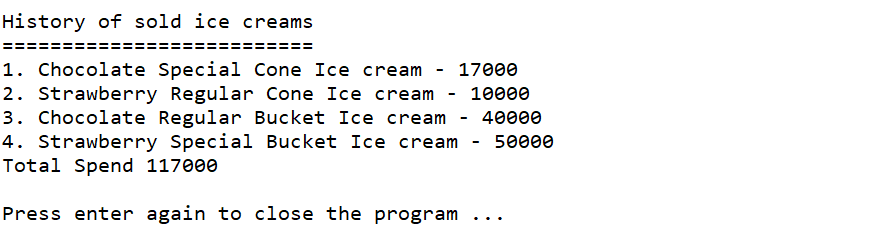


**Figure 7. Generate Special Bucket Ice Cream**

* 1. **Exit**

If the customer **choose 3**, then the program will:

* **Show the history of purchases** ice cream and the **total spend** by the user.
* And **close** the program.



**Figure 8. Show History of Purchases**