|  |  |
| --- | --- |
| **Assignment Case** |  |
| COMP6153001  Operating System |
| **Computer Science** | **O223-COMP6153-BD15-01** |
| ***Valid on*** *Odd Semester Year 2020/2021* | **Revision 00** |

1. Seluruh mahasiswa tidak diperkenankan untuk:

*All students are not allowed to:*

* + - Melihat sebagian atau seluruh jawaban mahasiswa lain,

*Seeing a part or the whole answer from other student*

* + - Menyadur sebagian maupun seluruh jawaban dari buku,

*Adapted a part or the whole answer from the book*

* + - Mendownload sebagian maupun seluruh jawaban dari internet,

*Downloading a part or the whole answer from the internet,*

* + - Mengerjakan soal yang tidak sesuai dengan tema yang ada di soal,

*Working with another theme which is not in accordance with the existing theme in the matter of the case,*

* + - Melakukan tindakan kecurangan lainnya,

*Committing other dishonest actions,*

* + - Secara sengaja maupun tidak sengaja melakukan segala tindakan kelalaian yang menyebabkan hasil karyanya berhasil dicontek oleh orang lain / kelompok lain.

*Accidentally or intentionally conduct any failure action that cause the results of the project was copied by someone else / other groups.*

1. Jika mahasiswa terbukti melakukan tindakan seperti yang dijelaskan butir 1 di atas, maka **nilai mahasiswa** yang melakukan kecurangan (menyontek maupun dicontek) akan di – **NOL** – kan.

*If the student is proved to the actions described in point 1 above, the score of the student which committed dishonest acts (cheating or being cheated) will be “Zero”*

1. Perhatikan jadwal pengumpulan jawaban, segala jenis pengumpulan jawaban di luar jadwal tidak dilayani.

*Pay attention to the submission schedule, all kinds of submission outside the schedule will not be accepted*

1. Persentase penilaiaan untuk matakuliah ini adalah sebagai berikut:

*Marking percentage for this subject is described as follows:*

|  |  |  |
| --- | --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* | **UAP**  *Final Exam* |
| 40% | - | 60% |

1. Software yang digunakan pada matakuliah ini adalah sebagai berikut:

*Software will be used in this subject are described as follows:*

|  |
| --- |
| **Software**  *Software* |
| VM Ubuntu Client 20.04  Java 8  Eclipse 2020.6  NachOS 5.0j |

## Ekstensi file yang harus disertakan dalam pengumpulan tugas mandiri dan uap untuk matakuliah ini adalah sebagai berikut:

*File extensions should be included in assignment and project collection for this subject are described as follows:*

|  |  |  |
| --- | --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* | **UAP**  *Final Exam* |
| DOCX, JAVA, CLASS | - | JAVA, CLASS |

## Soal

*Case*

**Bar Dash**

**Bar Dash** is a **game** that provides experience of owning a bar. As their DevOps, you are tasked to create a **bash command line** to do each of the following tasks:

1. Create a **Bar** directory in **home directory** using the following structure. (single execution)

<Home directory>

`-- Bar

|-- Boba

| |-- Milk Tea

| |-- Hazelnut Choco

|-- Tea

| |-- Green Tea

| |-- Utility

1. Create a file named "**recipe.txt**" inside "**Bar/Boba/Milk Tea**" and change the **access date** and **modify date** to **June 21, 2021 00:00:00**. (single execution)
2. From **home directory**, **search** for **file(s)** with the following criteria:
   * The filename **ends with** "**.txt**"
   * The file size is under 100 kibibytes
   * Inside the **Home** directory
3. Get the **process id** of process named **bash**.
4. Java Programming

**Bar Dash** also wants to **rebuild** their **sales** **system**. You are also tasked to create this program using **Java programming** language with following concepts:

1. Abstract Class

You need to design at least **three** classes, **one abstract** class, and **two concrete** classes. Abstract class consists of all **common** attributes and behavior that both of concrete class had. Concrete class consist of **specific** attribute and behavior that not common between the concrete classes

1. Encapsulation

To **hide** the data of a class from an **illegal** direct access, all of the attributes of the class must be **encapsulated** and will be accessed using an **accessor** and **mutator** that may perform validation before accessing the encapsulated attribute

1. Inheritance

All of the concrete class **must inherit all** attribute and behavior from the abstract class

1. Polymorphism

If the concrete class has **a specific implementation** of the inherited behavior (method) that **differ** from the abstract class, the concrete class can **override** or **overload** the behavior from the abstract class

The following are the specifications for the program:

* First, the program will **display main menu** consists of:
* Add Drink
* View Drink
* Exit



Figure 1. Menu

* If the user chooses **menu 1** (“**Add Drink**”), the program will prompt the user to input:
  + **Drink Type**. The user chooses either “**Tea**” or “**Boba**” (**case sensitive**).
  + **Drink Name**. The input must be **between** **5 and 25 characters** (**inclusive**).
  + **Drink Temperature**. The user chooses either “**Hot**” or “**Cold**” (**case sensitive**).
  + **Drink Sugar Amount**. The input must be **between** **1 and 5** (**inclusive**).
  + **Drink Base Price**. The input must be **between** **10000 and 150000** (**inclusive**).
  + If the user chooses **Tea** as the Drink Type, then:
    - Ask the user to input **Tea Leaves Weight**. The input must be **between 1 and 5** (**inclusive**).
  + If the user chooses **Boba** as the Drink Type, then:
    - Ask the user to input **Boba Topping**. The user chooses either “**Pearl**”, “**Grass Jelly**”, or “**Coffee Jelly**” (**case sensitive**).
  + Then the program will **insert** the **drink data** and **return to main menu**.

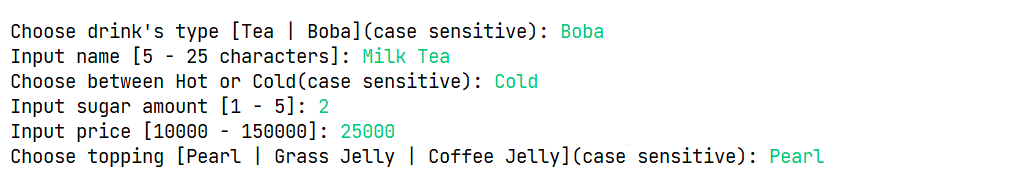


Figure 2. Add Drink (Boba)

* If the user chooses **menu 2** (“**View Drink**”):
  + If there is **no drink data**, the program will show **“no drink” message** and **return to main menu**.



*Figure 3. “No Drink” message*

* + **Otherwise**, the program will **display all drink data** that are stored on the list.
  + The final price of **Drink Type Tea** is **calculated** based on the following formula:

**Final Price = Drink Base Price +** (**Tea Leaves Weight \* 2000**)

* + The final price of **Drink Type Boba** is **calculated** based on the following formula:

**Final Price = Drink Base Price + Topping Price**

|  |  |
| --- | --- |
| **Topping** | **Price** |
| Pearl | 3000 |
| Grass Jelly | 4000 |
| Coffee Jelly | 5000 |

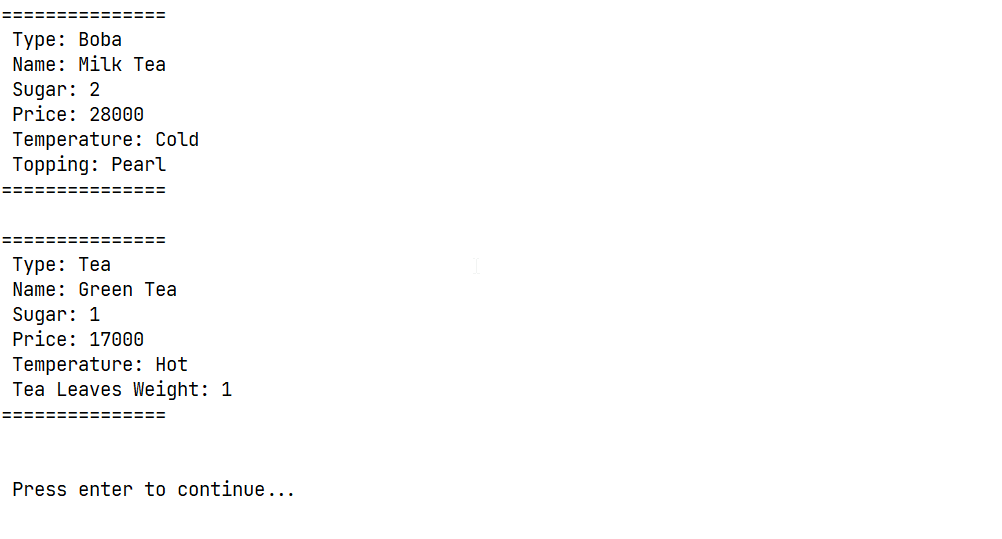


Figure 4. View Drink

* If the user chooses **menu 3** (“**Exit**”), the program will **exit**.