|  |  |
| --- | --- |
| **Assignment Case** |  |
| COMP6153  Operating System |
| **Computer Science** | **O213-COMP6153-FS05-01** |
| ***Valid on*** *Odd Semester Year 2020/2021* | **Revision 00** |

1. Seluruh mahasiswa tidak diperkenankan untuk:

*All students are not allowed to:*

* + 1. Melihat sebagian atau seluruh jawaban mahasiswa lain,

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

* + 1. Menyadur sebagian maupun seluruh jawaban dari buku,

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

* + 1. Mendownload sebagian maupun seluruh jawaban dari internet,

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

* + 1. 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,*

* + 1. Melakukan tindakan kecurangan lainnya,

*Committing other dishonest actions,*

* + 1. 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*

1. From the **home directory**, write a **command line** to create a directory based on the hierarchy below in a single execution! Note that folder names must be identical to the example below, including the spaces between words.

<Home directory>

`-- Coffee Store

|-- Coffee Types

| |-- Espresso

| |-- Macchiato

|-- Order List

|-- Stock List

| |-- Coffee Beans

* mkdir -p "Coffee Store"/"Coffee Types"/Espresso "Coffee Store"/"Coffee Types"/Macchiato "Coffee Store"/"Order List" "Coffee Store"/"Stock List"/"Coffee Beans"

1. From **Home Directory**, write a **command lin**e to create a new file named **Order 1.txt** inside the **Order List** folder, then **set permission** so **every user group** will only be able to **read** the file or in other words make the file **read-only** for **every group of users**.

* touch "Coffee Store"/"Order List"/"Order 1.txt"; chmod 444 "Coffee Store"/"Order List"/"Order 1.txt"

1. From **Home Directory,** write a **command line** to **search files** with the following criteria:

* **File Permission** is **read-only** for **every group of users**
* find "Coffee Store" -type f -perm -444

1. Write a **command line** to display the **PID, User, and Command** of **all processes** that were running/executed by the user **root**.

* ps -e -o pid,uid,comm -u root

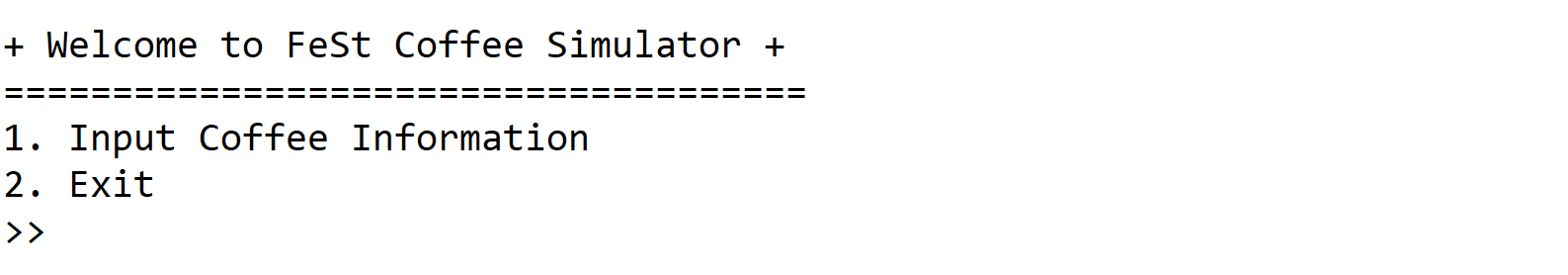
**Java Programming**

**FeSt Coffee Simulator**

**FeSt Coffee Simulator** is a Coffee Store Simulator that is used to simulate an order based on the user input type of coffee. As a programmer, you are asked to create a program to create this program for the grand opening of **FeSt Coffee Store**.

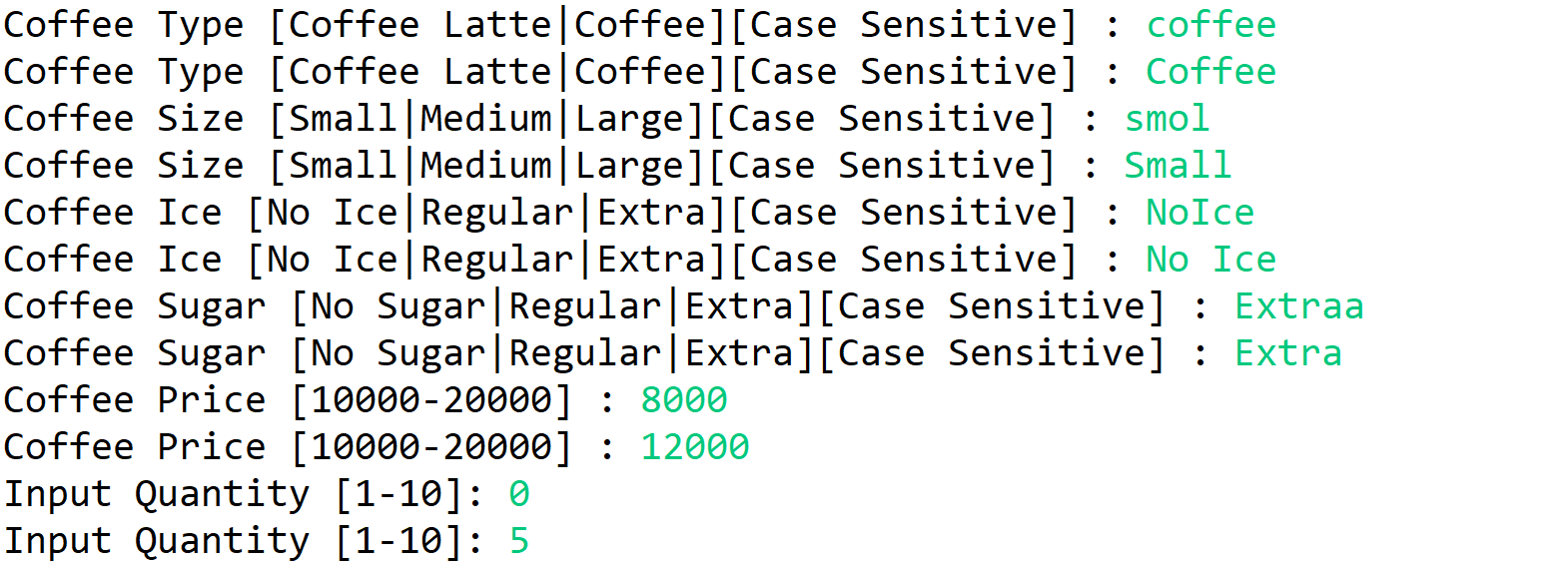
Here are the detail requirements of the application:

* The application must use **Object-Oriented Programming** concepts such as **Encapsulation**, **Inheritance**, and **Polymorphism**.There must be **Polymorphic Classes** and at least 1 **Overridden Method**.
* Create methods to calculate the **Total Price**, and **Simulate the Order**
* When the **program starts**, display a menu that asks if the user wants to **start the simulation (Input Coffee Information)** or to **Exit**.

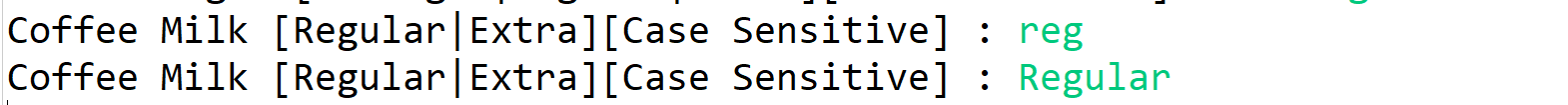


**Figure 1. Screenshot of the Main Menu**

* If the user selects the firstmenu, then:
* Ask the user to input **Coffee Type, Coffee Size, Amount of Ice, Amount of Sugar, Base Price,** and **Quantity** of the order. Validate the **Type** must be either **Coffee** or **Coffee Latte** (**case sensitive**).
* Ask the user to input **Coffee Size**. Validate the **Size** must be between “**Small”**, “**Medium”**, and “**Large”** (**case** **sensitive**).
* Ask the user to input the **Amount of Ice**. Validate the **Input** must be between “**No Ice”**, “**Regular”**, and “**Extra”** (**case** **sensitive**)
* Ask the user to input the **Amount of Sugar**. Validate the **Input** must be between “**No Sugar”**, “**Regular”**, and “**Extra”** (**case** **sensitive**)
* If the type of coffee is **Coffee Latte,** ask the user to input the **Amount of Milk**. Validate the **Input** must be either “**Regular**” or “**Extra**” (**case sensitive**)
* Ask the user to input **Base Price**. Validate the **Price** must be between **10.000** and **20.000** inclusively.
* Ask the user to input **Quantity**. Validate the **Quantity** **must be at least 1** and **at most 10**.

****

**Figure 2. Screenshot of input Coffee Information**



**Figure 3. Screenshot of input Amount of Milk**

* Determine the Coffee Price based on the following rules:

|  |  |
| --- | --- |
| **Size** | **Price** |
| Small | 0 |
| Medium | 4.000 |
| Large | 8.000 |

|  |  |
| --- | --- |
| **Sugar** | **Price** |
| No Sugar | 0 |
| Regular | 2.000 |
| Extra | 4.000 |

|  |  |
| --- | --- |
| **Ice** | **Price** |
| No Ice | 0 |
| Regular | 4.000 |
| Extra | 6.000 |

|  |  |
| --- | --- |
| **Milk** | **Price** |
| Regular | 5.000 |
| Extra | 7.000 |

* Determine the **Coffee Price** with the following formula:

**Coffee Price = Base Price + Size Price + Ice Price + Sugar Price**

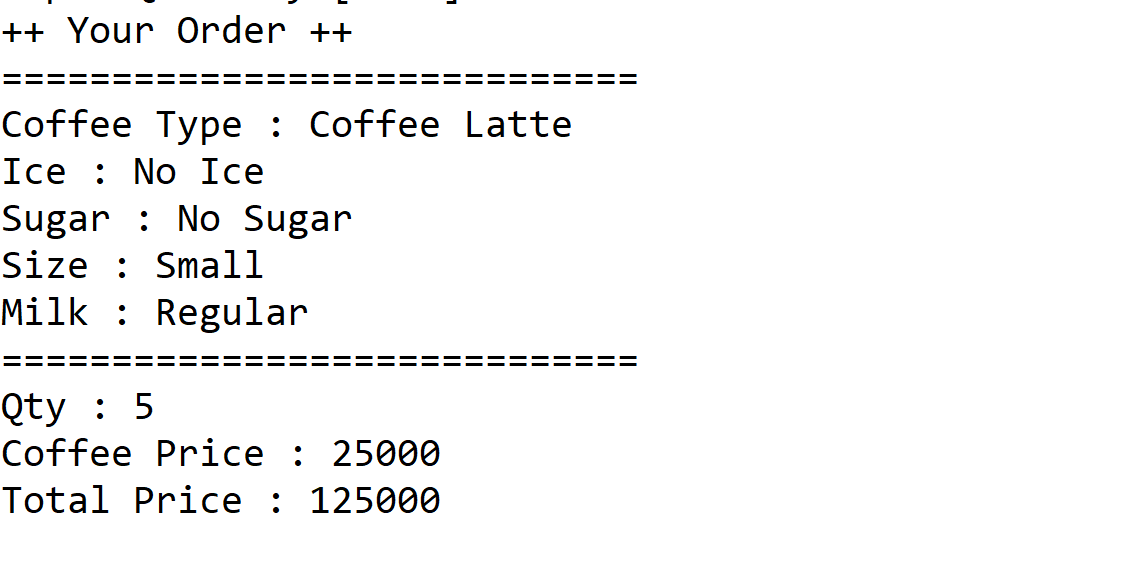
If the **Coffee Type** is **Coffee Latte**, then add Milk Price into the calculation:

**Coffee Price = Base Price + Size Price + Ice Price + Sugar Price + Milk Price**

* Determine the **Total Price** by multiplying the **Coffee Price** and the **Quantity**:

**Total Price = Coffee Price \* Quantity**

* Display **Coffee Type**, **Coffee Size**, **Coffee Ice**, **Coffee Sugar**, **Quantity**, **Coffee Price,** and **Total** **Price** of the inputted order. If the **Coffee Type** is **Coffee Latte**, display the **Milk Detail** as well.



**Figure 4. Screenshot of Coffee Order**

* After the order is simulated, return the user to **Main Menu**.