|  |  |
| --- | --- |
| **Assignment Case** |  |
| COMP6708016 Object Oriented Programming |
| **Computer Science** | **E223-COMP6708016-LC03113-01** |
| ***Valid on*** *Even Semester Year 2021/2022* | **Revision 00** |

## Soal

*Case*

**Comp Shop 31**

**Comp Shop 31** is an application where you can build your own pc with the available pc parts and user can also view and delete the transaction. This application will allow the customer to **buy PC** by choosing the computer parts. This application can also allow the user to check the **Transaction History** and let them to **Remove Transaction** from the transaction list. You must create the program using **JAVA** **Programming** **Language** with **Object** **Oriented** **Programming** concept such as **Encapsulation**, **Composition**, and **Aggregation**. The requirements for the application are listed below:

* In the beginning, the program will show **four menus**:

1. **Buy PC**
2. **View Transaction**
3. **Remove Transaction**
4. **Exit**

A picture containing graphical user interface

Description automatically generated

* If the user chose **menu 1**, then:
  + Ask the user to input **processor brand** which must be **either ‘Intel’ or ‘AMD’ (case sensitive)**.
  + Ask the user to input **processor**
    - If user chose ‘**Intel’** as the **processor brand**.Theusermustchoose **either ‘12600K’, ‘12700K’ or ‘12900K’ (case sensitive)**.
    - If user chose ‘**Intel’** as the **processor brand**.Theusermustchoose **either ‘5600X’, ‘5700X’ or ‘5950X’ (case sensitive)**.
  + Ask user to choose **GPU**. The user must choose **either ‘RTX 2060’, ‘RTX 3070’ or ‘GTX 1080’ (case sensitive)**.
  + Ask the user to input **ram size** which **must be either ‘8’, ‘16’ or ‘32’ GB.**
  + Ask the user to input **storage size** which **must be either ‘500’, ‘1024’ or ‘2048’ GB.**
  + Ask the user to input **Customer Name which cannot be empty**.
  + Ask the user to input **Shipping Address which must be start with 'jl.<<space>>’**
  + For the Transaction ID, **generate the ID** using following format:

**Transaction ID’s format**: XXYYY

Where:

X: **Random** **uppercase** **letter** [**A-Z**]

Y: **Random** **number** [**0-9**]

Example: **BK185**

* + For the item price, it will be automatically **generated** like below:

|  |  |
| --- | --- |
| Processor Intel | Price |
| 12600K | 1000 |
| 12700K | 1100 |
| 12900K | 1200 |

|  |  |
| --- | --- |
| Processor AMD | Price |
| 5600X | 800 |
| 5700X | 900 |
| 5950X | 1000 |

|  |  |
| --- | --- |
| GPU | Price |
| RTX 2060 | 2860 |
| RTX 3070 | 5650 |
| GTX 1080 | 1200 |

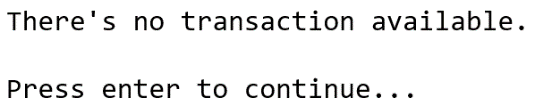
* + For the **price** formula:

**Price = (1.1 \* (Processor Price + GPU Price)) + 100**

Text

Description automatically generated

* + Then, the program will **add** the **transaction** into **transaction** **list** (**ArrayList / Vector / Array**) and return to **main menu**.
* If the user chose **menu 2**, then:
  + If there is **no** **transaction** in the **list**, show **error** **message**. Then return to **main menu**.



* + **Otherwise**, show all **transactions** including its **details**.

Table

Description automatically generated

* If the user chose **menu 3**, then:
  + If there is **no** **transaction** in the **list**, show **error** **message**. Then return to **main menu**.

Text

Description automatically generated

* + **Otherwise**, show all **transactions** including its **details**.
  + Then, the program will ask user which **transaction ID** to be **removed**. If user inputted ‘**0**’ it will return to **main menu**. Validate the inputted **transaction** **ID** must **matched** with **transaction** in the **list**.

Table

Description automatically generated

* + **Remove** the **matched** **transaction** from the list. Then, return to **main menu**.

Table

Description automatically generated

* If the user chose **menu 4**, then exit the program.

**If you need any assistance, kindly ask your assistants for help.**