|  |  |
| --- | --- |
| **Assignment Case** |  |
| COMP6048  Data Structures |
| **Computer Science** | **C193-COMP6048-FT03602-01** |
| ***Valid on*** *Even Semester Year 2020/2021* | **Revision 00** |

1. Seluruh mahasiswa tidak diperkenankan untuk:

*All students are not allowed to:*

* + - Berdiskusi dan/atau bekerja sama dengan mahasiswa lainnya

*Discuss and/or work together with other student participants*

* + - Melihat sebagian atau seluruh jawaban mahasiswa lain

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

* + - Membuka dan menyalin dari **BUKU** atau **CATATAN**, **VIDEO** dari pengajar (recording kelas, VBL, Youtube, dsb) dan **REFERENSI** lainnya

*Open and copy from any resources such as notes, videos (class recording, VBL, Youtube, etc) and other references*

* + - Membuka dan menyalin jawaban dari internet (google, stackoverflow, dsb)

*Open and copy answer from the internet (google, stackoverflow, etc)*

* + - 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. Bila Anda tidak membaca peraturan ini, maka Anda dianggap telah membaca dan menyetujuinya

*If you have missed to read these regulations, so you are considered to have read and agreed on it*

1. Persentase penilaiaan untuk matakuliah ini adalah sebagai berikut:

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

|  |  |
| --- | --- |
| **Tugas Mandiri**  *Assignment* | **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* |
| Dev-C++ 5.11 |

## 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 final exam collection for this subject are described as follows:*

|  |  |
| --- | --- |
| **Tugas Mandiri**  *Assignment* | **UAP**  *Final Exam* |
| CPP | CPP |

**Soal**

*Case*

**Nake Warehouse**

**Nake Warehouse** is a famous warehouse that sell any sport things in Tangerang. **Nake Warehouse** is a big company, so to easily manage their products, **Nake Warehouse** wanted a program that is easy to maintenance. You as a programmer and a close friend of **Nake Warehouse**’s CEO are asked to create program that can maintain product data using **C language** and **Hash Table** data structure. The program that will be created must be following the below requirements.

* The program will use **Hash Table** data structure to store products data with the following details:
  + The hash table’s size is 55
  + To determine a product’s index in the hash table, use the following **hash function**

**Key = (X-1) % Y**

**Where:**

Key : hash table index for the product

X : the last 2 digits of the product ID

Y : size of the hash table (55)

**Example:**

Item Id : NKW05

Size : 55

Key : (05 – 1) % 55

: 4

Then the item data will store at index 4 of hash table

Figure 1 Hash function

* + Use **chaining** technique to handle collisions
* The program **with three main menus:**

1. **Insert New Product**
2. **Delete Product**
3. **Exit**

* Before showing main menu, the program will also **show all products stored** in the program
  + **If** there is **no products data** stored, the program will **show message** “**There is No Data !”**



Figure 2 Main menu (no data)

* + Otherwise, the program will **list all products** and shows them with the following details

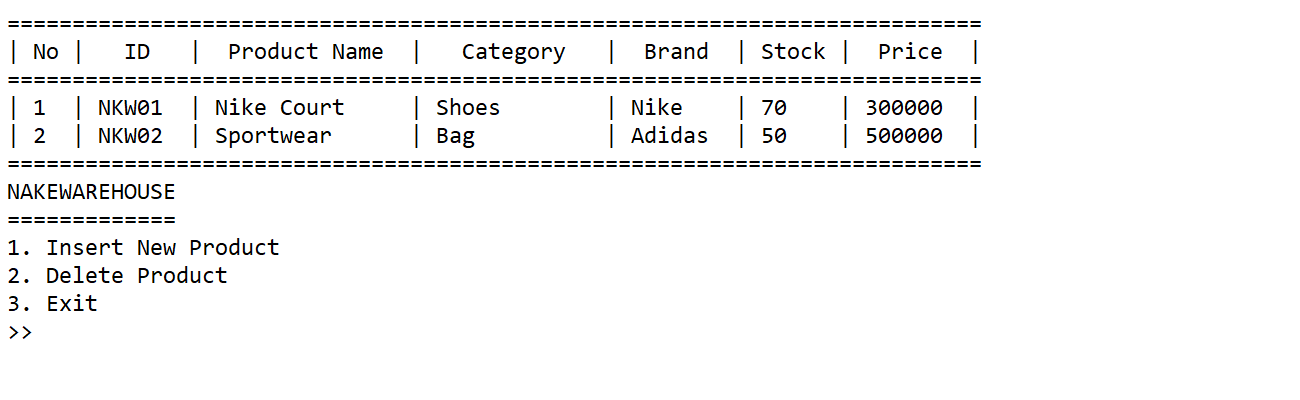


Figure 3 Main menu (with data)

* If the user choose **menu 1** (**Insert New Product**), then:
  + The program will ask the user to input the following fields:
    - **Product name**, whose **character length** must be **between 5 and 24**
    - **Product category**, whose **value** must be **between** “**Shoes**”or “**Bag**” (**case sensitive**)
    - **Product brand**, whose **value** must be **between** “**Nike**” or “**Adidas**” (**case sensitive**)
    - **Product stock**, whose **value** must be **between** **50 - 150 pcs**
    - **Product price**, whose **value** must be **between** **200000 and 1000000**

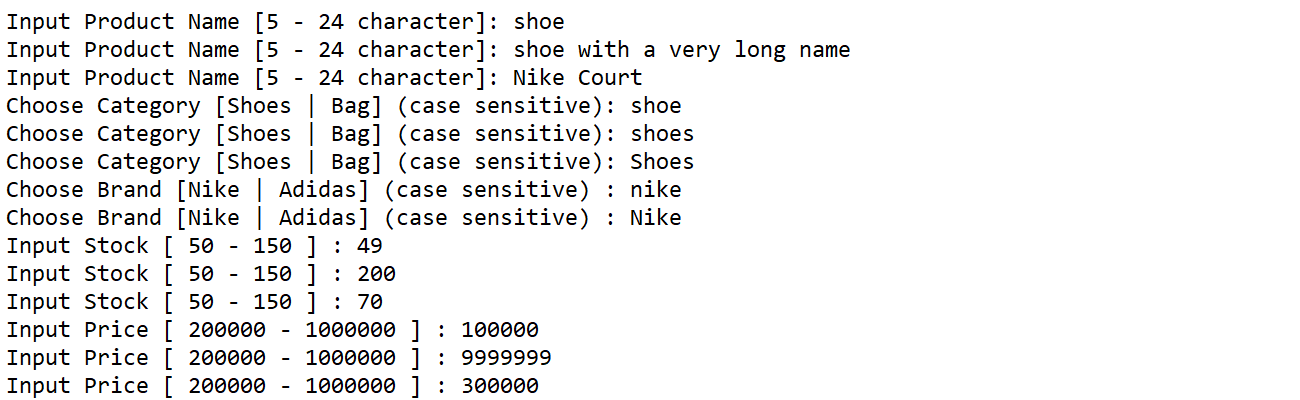


Figure 4 Insert product validation

* + After obtained all the product’s data, the program will **generate** a **product ID** for the inputted product following the format:

**NKWXX**

**Where**:

XX : the last 2 digits of the last item id added by 1

**Example**:

The last item id is **NKW01**

Then the new generated item id is **NKW02**

**Note:**

The first inserted products’ ID is **NKW01**

Figure 5 Product ID format

* + The program will **store the** newly inputted **data** and **return to main menu**
* If the user choose **menu 2** (**Delete Product**), then:
  + **If** there is **no products data** stored, the program will **show message** “**There is No Data !”** and **return to main menu**



Figure 6 Delete product (no data)

* + Otherwise, the program will **show the data** and ask user to input the **Product ID** that they want to **delete**. **Validate** the **Product ID** inputted **exist** in the hash table **(case insensitive)**

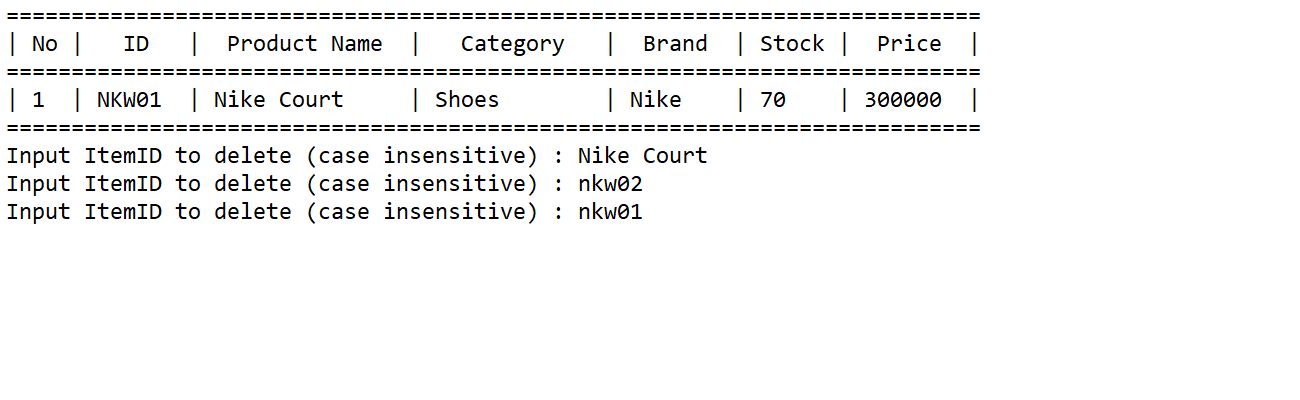


Figure 7 Delete product validation

* + The program will then find and **delete** the **chosen** **product data** and **return to main menu**
* If the user choose **menu 3** (**Exit**), then the program will **be closed**

**Please run the EXE file to get more detail about the application**