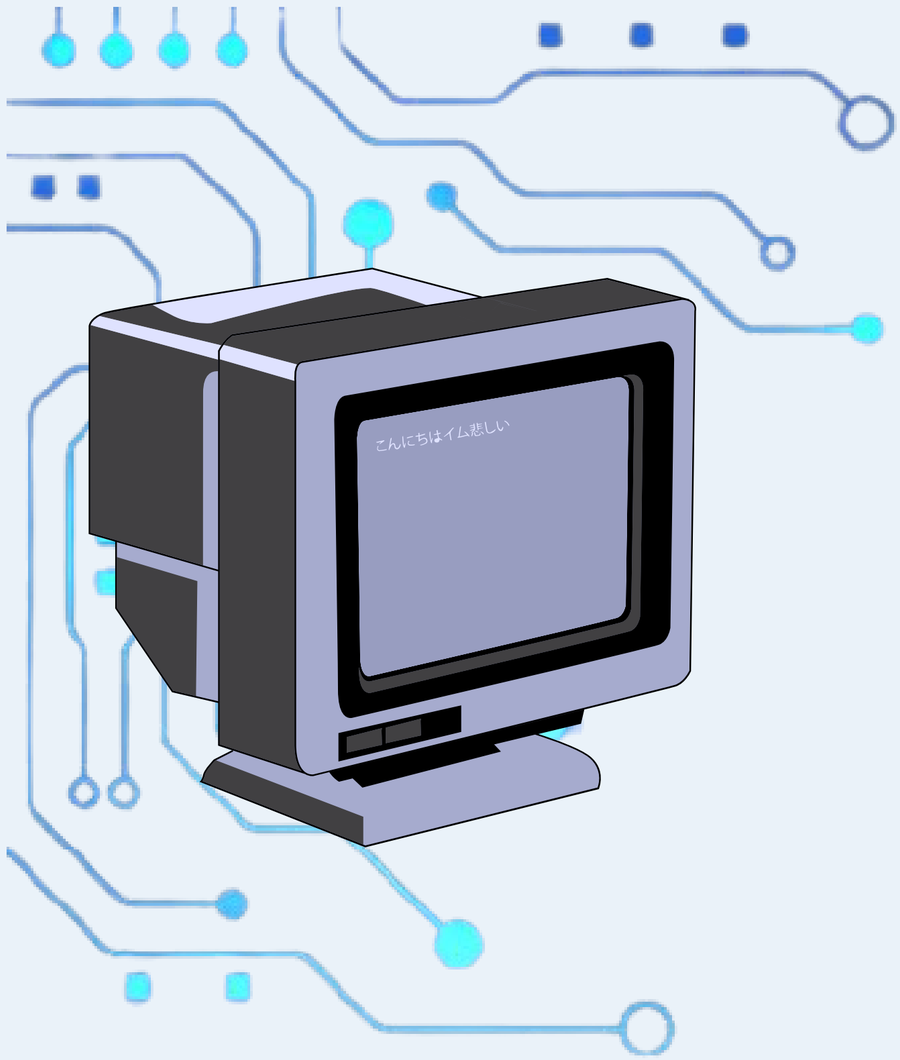


Inventory Management System of Computer Stock

Content

- Introduction to Inventory Management system
- Importance of Inventory Management
- Inventory Management System of computer stock
- Overview of code
- Menu options
- Conclusion

Introduction to Inventory Management

- Inventory management is the practice of efficiently overseeing and controlling a company's inventory of goods, products, or materials.

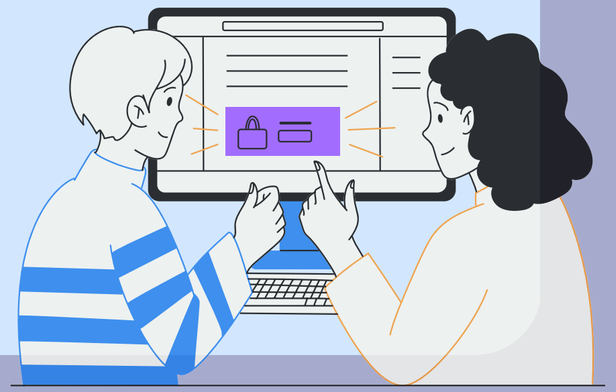
- It involves monitoring the quantities, product name, and price of inventory items etc to ensure that a business has the right products available at the right time while minimizing carrying costs and preventing overstock or stockouts.

- Effective inventory management is essential for businesses to balance the costs of holding inventory with the need to meet customer demand. It involves a combination of data analysis, forecasting, technology, and process optimization to maintain the right level of inventory and ensure the overall success of the business.

Importance of Inventory Management

- Inventory includes all the items a business uses to produce its goods or services, raw materials, work-in-progress items, and finished products.

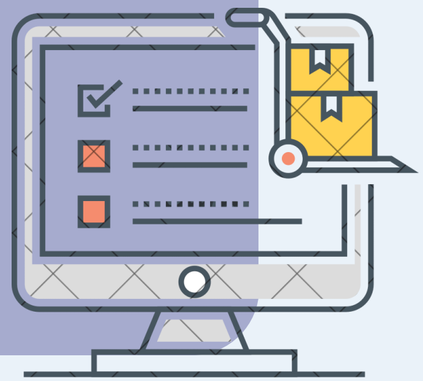
- Proper inventory management is crucial for the smooth operation of various business functions, including manufacturing, sales, and customer service.



Inventory Management System of computer stock

- C++ code provides a basic framework for managing an inventory of computer items in a store, allowing users to perform essential inventory operations on the store's inventory, including inserting items, deleting items, searching for items, updating item information, and displaying the entire inventory.

- It serves as a starting point and can be extended and enhanced to meet specific requirements and handle various edge cases.



Overview of code

Header Files and Namespace:
The code includes two standard C++ header files: <iostream> for input and output operations and <string.h> for string manipulation.
It uses the using namespace std; directive to simplify the use of standard C++ objects and functions

Data Structures:
The code defines a struct named 'itemEntry' to represent an item's information, including its unit price, number of copies, product ID, name, and company.

Class: Store:

- The core functionality of the inventory management system is encapsulated in a class named Store.
- Public members of the class:
  - int numItem: Stores the number of items in the inventory.
  - itemEntry database[100]: An array of itemEntry structures to store the inventory data.
- The class includes a constructor Store() that initializes numItem to 0.

Member Functions of the Store Class:

void insertItem(char itemName[], char company[], int pid, int c, float p): Inserts a new item into the inventory by copying the provided information.

void deleteItem(char itemName[], int pid): Decrements the number of copies of an item by 1.

itemEntry *searchItem(char itemName[], int pid): Searches for an item in the inventory based on its name and product ID and returns a pointer to the item if found.

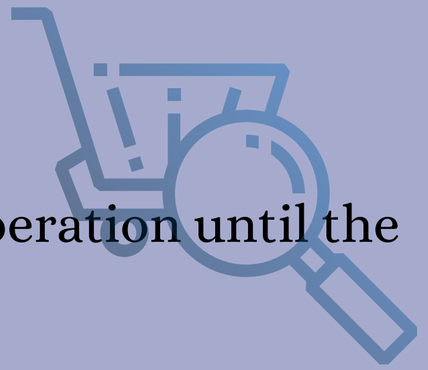
void updateItem(char itemName[], int pid, int total, float price): Updates the number of copies and unit price of an existing item.

void displayInventory(): Displays the entire inventory.

Main Function:

The main function creates an instance of the 'Store' class named 'sto'.

It uses a do-while loop to display a menu to the user and execute the chosen operation until the user chooses to exit.



User Input Handling:
The program uses cin and cin.getline() for user input, allowing the user to enter item names and company names with spaces.
cin.ignore() is used to clear any newline characters left in the input buffer.

Output:
The program provides appropriate feedback messages for each operation, indicating whether an operation was successful or if an item was not found.

Menu Options:

The menu provides six options:

1. Insert Item
2. Delete Item
3. Search Item
4. Update Item
5. Display Item Inventory
6. Exit

SOURCE CODE:

D:\C++\inventory.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

Project Classes Debug | oops exps.cpp | inventory.cpp

```cpp
#include <iostream>
#include <string.h>

using namespace std;

struct itemEntry
{
    float unitPrice;
    int copies, product_id;
    char name[30];
    char company[30];
};

class Store
{
public:
    int numItem;
    itemEntry database[100];

    Store()
    {
        numItem = 0;
    }

    void insertItem(char itemName[], char company[], int pid, int c, float p);
    void deleteItem(char itemName[], int pid);
    itemEntry *searchItem(char itemName[], int pid);
    void updateItem(char itemName[], int pid, int total, float price);
    void displayInventory();
};

void Store::insertItem(char itemName[], char company[], int pid, int c, float p)
{
    strcpy(database[numItem].name, itemName);
    strcpy(database[numItem].company, company);
    database[numItem].product_id = pid;
    database[numItem].copies = c;
```

Compiler Resources Compile Log Debug Find Results

Line: 1 Col: 1 Sel: 0 Lines: 178 Length: 5773 Insert Done parsing in 0.281 seconds

SOURCE CODE:

D:\C++\inventory.cpp - Dev-C++ 5.11

```cpp
void Store::insertItem(char itemName[], char company[], int pid, int c, float p)
{
    strcpy(database[numItem].name, itemName);
    strcpy(database[numItem].company, company);
    database[numItem].product_id = pid;
    database[numItem].copies = c;
    database[numItem].unitPrice = p;
    cout << "\n\t\t ITEM INSERTED SUCCESSFULLY\n";
    numItem++;
}

void Store::deleteItem(char itemName[], int pid)
{
    for (int i = 0; i < numItem; i++)
    {
        if ((strcmp(itemName, database[i].name) == 0) && (database[i].product_id == pid))
        {
            database[i].copies--;
            cout << "\n\t\t ITEM DELETED SUCCESSFULLY\n";
            return;
        }
    }
    cout << "\n\t\t ITEM NOT FOUND\n";
}

itemEntry *Store::searchItem(char itemName[], int pid)
{
    for (int i = 0; i < numItem; i++)
    {
        if ((strcmp(itemName, database[i].name) == 0) && (database[i].product_id == pid))
            return &database[i];
    }
    return NULL;
}


void Store::updateItem(char itemName[], int pid, int total, float price)
```

SOURCE CODE:

D:\C++\inventory.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug | oops exps.cpp | [*] inventory.cpp

```cpp
void Store::updateItem(char itemName[], int pid, int total, float price)
{
    itemEntry *item = searchItem(itemName, pid);
    if (item != NULL)
    {
        item->copies += total;
        item->unitPrice += price;
    }
    else
    {
        cout << "\n\t\t ITEM NOT FOUND\n";
    }
}

void Store::displayInventory()
{
    cout << "\n\t------------>INVENTORY<--------------\n";
    for (int i = 0; i < numItem; i++)
    {
        cout << "\n\tItem " << i + 1 << ":\n";
        cout << "\t\tName: " << database[i].name << "\n";
        cout << "\t\tCompany: " << database[i].company << "\n";
        cout << "\t\tProduct ID: " << database[i].product_id << "\n";
        cout << "\t\tCopies: " << database[i].copies << "\n";
        cout << "\t\tUnit Price: " << database[i].unitPrice << "\n";
    }
}

int main()
{
    Store sto;
    char name[30], company[30];
    int product_id, copies, unit_price, option;

    do
    {
        cout << "\n\t------------COMPUTER STOCK STORE<--------------";
```

Compiler Resources Compile Log Debug Find Results

Line: 68 Col: 2 Sel: 0 Lines: 179 Length: 5775 Insert Done parsing in 0.281 seconds

32°C Haze Search ENG IN 12:10 24-10-2023

SOURCE CODE:

D:\C++\inventory.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

Project Classes Debug | oops exps.cpp | [*] inventory.cpp

```cpp
    cout << "\n\t-------------COMPUTER STOCK STORE<--------------";
    cout << "\n\t\t-------------->MENU<------------";
    cout << "\n\t\t\t    1 FOR INSERT ITEM";
    cout << "\n\t\t\t    2 FOR DELETE ITEM";
    cout << "\n\t\t\t    3 FOR SEARCH ITEM ";
    cout << "\n\t\t\t    4 FOR UPDATE ITEM ";
    cout << "\n\t\t\t    5 FOR DISPLAY ITEM--------------------------------8";
    cout << "\n\t\t\t    6 FOR EXIT";
    cout << "\n\t\t  ENTER YOUR CHOICE : ";
    cin >> option;

    switch (option)
    {
    case 1:
        cin.ignore(); // Ignore newline character left in the buffer
        cout << "\n\t\t\t ENTER NAME OF COMPUTER MODEL : ";
        cin.getline(name, 80);
        cout << "\n\t\t\t COMPANY: ";
        cin.getline(company, 80);
        cout << "\n\t\t\t ENTER PRODUCT ID : ";
        cin >> product_id;
        cout << "\n\t\t\t NO. OF COPIES : ";
        cin >> copies;
        cout << "\n\t\t\t UNIT PRICE PER ITEM : ";
        cin >> unit_price;
        sto.insertItem(name, company, product_id, copies, unit_price);
        break;
    case 2:
        cin.ignore(); // Ignore newline character left in the buffer
        cout << "\n\t\t\t ENTER NAME OF COMPUTER MODEL : ";
        cin.getline(name, 80);
        cout << "\n\t\t\t ENTER PRODUCT ID : ";
        cin >> product_id;
        sto.deleteItem(name, product_id);
        break;
    case 3:
        cin.ignore(); // Ignore newline character left in the buffer
```

Compiler Resources Compile Log Debug Find Results

Line: 68 Col: 2 Sel: 0 Lines: 179 Length: 5775 Insert Done parsing in 0.281 seconds

SOURCE CODE:

D:\C++\inventory.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

Project Classes Debug | oops exps.cpp | [*] inventory.cpp

```cpp
        case 3:
            cin.ignore(); // Ignore newline character left in the buffer
            cout << "\n\t\t\t ENTER NAME OF COMPUTER MODEL : ";
            cin.getline(name, 80);
            cout << "\n\t\t\t ENTER PRODUCT ID : ";
            cin >> product_id;
            itemEntry *test;
            test = sto.searchItem(name, product_id);
            if (test != NULL)
            {
                cout << "\n\t------------->SEARCHING RESULT<--------------";
                cout << "\n\t\t\t   ITEM FOUND"
                     << "\n\t\t\t NAME OF THE COMPUTER MODEL :" << test->name << "\n\t\t\t COMPANY NAME :"
                     << test->company << "\n\t\t\t PRODUCT ID:" << test->product_id
                     << "\n\t\t\t NUMBER OF COPIES AVAILABLE:" << test->copies
                     << "\n\t\t\t UNIT PRICES PER ITEM:" << test->unitPrice;
            }
            else
                cout << "\n\t\t\t ITEM NOT FOUND";
            break;
        case 4:
            cin.ignore(); // Ignore newline character left in the buffer
            cout << "\n\t\t\t ENTER NAME OF COMPUTER MODEL : ";
            cin.getline(name, 80);
            cout << "\n\t\t\t ENTER PRODUCT ID : ";
            cin >> product_id;
            cout << "\n\t\t\t ENTER TOTAL NEW ENTRY : ";
            cin >> copies;
            cout << "\n\t\t\t ENTER NEW PRICE : ";
            cin >> unit_price;
            sto.updateItem(name, product_id, copies, unit_price);
            break;
        case 5:
            sto.displayInventory();
            break;
        }
    } while (option != 6);
```

Compiler Resources Compile Log Debug Find Results

Line: 68 Col: 2 Sel: 0 Lines: 179 Length: 5775 Insert Done parsing in 0.281 seconds

32°C Haze Search ENG IN 12:12 24-10-2023

SOURCE CODE:

D:\C++\inventory.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug | oops exps.cpp | [*] inventory.cpp

```
            cout << "\n\t\t\t ENTER PRODUCT ID : ";
            cin >> product_id;
            itemEntry *test;
            test = sto.searchItem(name, product_id);
            if (test != NULL)
            {
                cout << "\n\t------------->SEARCHING RESULT<--------------";
                cout << "\n\t\t\t   ITEM FOUND"
                     << "\n\t\t\t NAME OF THE COMPUTER MODEL :" << test->name << "\n\t\t\t COMPANY NAME :"
                     << test->company << "\n\t\t\t PRODUCT ID:" << test->product_id
                     << "\n\t\t\t NUMBER OF COPIES AVAILABLE:" << test->copies
                     << "\n\t\t\t UNIT PRICES PER ITEM:" << test->unitPrice;
            }
            else
                cout << "\n\t\t\t ITEM NOT FOUND";
            break;
        case 4:
            cin.ignore(); // Ignore newline character left in the buffer
            cout << "\n\t\t\t ENTER NAME OF COMPUTER MODEL : ";
            cin.getline(name, 80);
            cout << "\n\t\t\t ENTER PRODUCT ID : ";
            cin >> product_id;
            cout << "\n\t\t\t ENTER TOTAL NEW ENTRY : ";
            cin >> copies;
            cout << "\n\t\t\t ENTER NEW PRICE : ";
            cin >> unit_price;
            sto.updateItem(name, product_id, copies, unit_price);
            break;
        case 5:
            sto.displayInventory();
            break;
        }
    } while (option != 6);

    return 0;
}
```

Compiler Resources Compile Log Debug Find Results

Line: 68 Col: 2 Sel: 0 Lines: 179 Length: 5775 Insert Done parsing in 0.281 seconds

32°C Haze Search ENG IN 12:13 24-10-2023

Introduction to Inventory Management

- Inventory management is the practice of efficiently overseeing and controlling a company's inventory of goods, products, or materials.

- It involves monitoring the quantities, product name, and price of inventory items etc to ensure that a business has the right products available at the right time while minimizing carrying costs and preventing overstock or stockouts.

- Effective inventory management is essential for businesses to balance the costs of holding inventory with the need to meet customer demand. It involves a combination of data analysis, forecasting, technology, and process optimization to maintain the right level of inventory and ensure the overall success of the business.

Output:

INSERT ITEM:

```
-------------COMPUTER STOCK STORE<--------------
        --------------->MENU<------------
                    1 FOR INSERT ITEM
                    2 FOR DELETE ITEM
                    3 FOR SEARCH ITEM
                    4 FOR UPDATE ITEM
                    5 FOR DISPLAY ITEM---------------------------------8
                    6 FOR EXIT
            ENTER YOUR CHOICE : 1

                    ENTER NAME OF COMPUTER MODEL : dell

                    COMPANY: dell

                    ENTER PRODUCT ID : 12345

                    NO. OF COPIES : 6

                    UNIT PRICE PER ITEM : 56000

           ITEM INSERTED SUCCESSFULLY
```

Output:

DELETE ITEM:

```
-------------COMPUTER STOCK STORE<--------------
        -------------->MENU<------------
                    1 FOR INSERT ITEM
                    2 FOR DELETE ITEM
                    3 FOR SEARCH ITEM
                    4 FOR UPDATE ITEM
                    5 FOR DISPLAY ITEM----------------------------------
                    6 FOR EXIT
            ENTER YOUR CHOICE : 2

                   ENTER NAME OF COMPUTER MODEL : dell

                   ENTER PRODUCT ID : 12345

           ITEM DELETED SUCCESSFULLY
```

Output:

SEARCH ITEM:

```
-------------COMPUTER STOCK STORE<--------------
        --------------->MENU<------------
                    1 FOR INSERT ITEM
                    2 FOR DELETE ITEM
                    3 FOR SEARCH ITEM
                    4 FOR UPDATE ITEM
                    5 FOR DISPLAY ITEM--------------------------------
                    6 FOR EXIT
          ENTER YOUR CHOICE : 3

                  ENTER NAME OF COMPUTER MODEL : HP

                  ENTER PRODUCT ID : 123456

-------------->SEARCHING RESULT<--------------
                    ITEM FOUND
                  NAME OF THE COMPUTER MODEL :HP
                  COMPANY NAME :HP
                  PRODUCT ID:123456
                  NUMBER OF COPIES AVAILABLE:5
                  UNIT PRICES PER ITEM:60000
```

Output:

UPDATE ITEM:

```
-------------COMPUTER STOCK STORE<--------------
        --------------->MENU<------------
                    1 FOR INSERT ITEM
                    2 FOR DELETE ITEM
                    3 FOR SEARCH ITEM
                    4 FOR UPDATE ITEM
                    5 FOR DISPLAY ITEM--------------------------------
                    6 FOR EXIT
          ENTER YOUR CHOICE : 4

                 ENTER NAME OF COMPUTER MODEL : hp

                 ENTER PRODUCT ID : 123456

                 ENTER TOTAL NEW ENTRY : 5

                 ENTER NEW PRICE : 60000
```

Output:

DISPLAY ITEM:

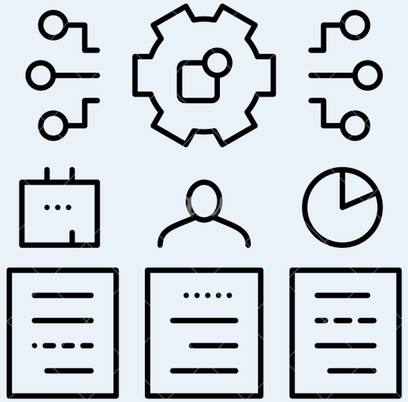
```
-------------COMPUTER STOCK STORE<--------------
        --------------->MENU<------------
                    1 FOR INSERT ITEM
                    2 FOR DELETE ITEM
                    3 FOR SEARCH ITEM
                    4 FOR UPDATE ITEM
                    5 FOR DISPLAY ITEM----------------------------------
                    6 FOR EXIT
          ENTER YOUR CHOICE : 5

------------->INVENTORY<--------------

Item 1:
        Name: HP
        Company: HP
        Product ID: 123456
        Copies: 10
        Unit Price: 120000
```

Conclusion:-

- The provided C++ code represents a basic inventory management system for a computer stock store.

- While the code offers functionality for inserting, deleting, searching, updating, and displaying items in the inventory, there are areas where it can be improved and extended to make it more robust and user-friendly.

- Our code serves as a starting point for an inventory management system but should be extended and refined to meet specific business requirements and handle real-world scenarios effectively.



THANK YOU

Unnati Pimple - 46
Tanisha Purohit - 47
Shruti Rathod - 49
Surabhi Raut - 50