***Muhammad Kaleem***

**56614**

**Lab task 6**

#include<iostream>

using namespace std;

class Item {

private:

int id;

string name;

string type;

int price;

int quantity;

Item\* next;

public:

Item(int id, string name, string type, int price, int quantity)

{

this->id = id;

this->name = name;

this->type = type;

this->price = price;

this->quantity = quantity;

this->next = NULL;

}

void setID(int id)

{

this->id = id;

}

void setName(string name)

{

this->name = name;

}

void setType(string type)

{

this->type = type;

}

void setPrice(int price)

{

this->price = price;

}

void setQuantity(int quantity)

{

this->quantity = quantity;

}

void setNext(Item\* next)

{

this->next = next;

}

int getID()

{

return id;

}

string getName()

{

return name;

}

string getType()

{

return type;

}

int getPrice()

{

return price;

}

int getQuantity()

{

return quantity;

}

Item\* getNext()

{

return next;

}

};

class Inventory {

private:

Item\* head = NULL;

public:

Inventory()

{

this->head = NULL;

}

void add\_item()

{

int id;

string name;

string type;

int price;

int quantity;

cout << "Enter Id : ";

cin >> id;

cout << "Enter name : ";

cin >> name;

cout << "Enter type : ";

cin >> type;

cout << "Enter price : ";

cin >> price;

cout << "Enter quantity : ";

cin >> quantity;

Item\* item = new Item(id, name, type, price, quantity);

if (head == NULL)

{

head = item;

return;

}

Item\* temp = head;

while (temp->getNext() != NULL)

{

temp = temp->getNext();

}

temp->setNext(item);

}

void drop\_item()

{

int id;

cout << "Enter id : ";

cin >> id;

if (head == NULL)

{

cout << "Inventory is empty." << endl;

return;

}

Item\* temp = head;

Item\* previous = NULL;

while (temp->getNext() != NULL)

{

if (temp->getID() == id)

{

if (previous == NULL)

{

head = temp->getNext();

}

else

{

previous->setNext(temp->getNext());

}

delete temp;

return;

}

previous = temp;

temp = temp->getNext();

}

}

void update\_item()

{

int id;

cout << "Enter id of the item to update: ";

cin >> id;

if (head == NULL)

{

cout << "Inventory is empty." << endl;

return;

}

Item\* temp = head;

while (temp != NULL)

{

if (temp->getID() == id)

{

int newId;

string newName, newType;

int newPrice, newQuantity;

cout << "Enter new Id: ";

cin >> newId;

cout << "Enter new name: ";

cin >> newName;

cout << "Enter new type: ";

cin >> newType;

cout << "Enter new price: ";

cin >> newPrice;

cout << "Enter new quantity: ";

cin >> newQuantity;

temp->setID(newId);

temp->setName(newName);

temp->setType(newType);

temp->setPrice(newPrice);

temp->setQuantity(newQuantity);

cout << "Item updated successfully." << endl;

return;

}

temp = temp->getNext();

}

cout << "Item with ID " << id << " not found." << endl;

}

};

int main()

{

Inventory INV;

int choice;

do

{

cout << " Inventory Management System \n";

cout << " 1) Add item \n";

cout << " 2) Delete item \n";

cout << " 3) Update item \n";

cout << "Enter a choice :";

cin >> choice;

switch (choice)

{

case 1:

INV.add\_item();

break;

case 2:

INV.drop\_item();

break;

case 3:

INV.update\_item();

break;

}

} while (true);

}

**Output:**

