**LAB 04**

**QUESTION 1:**

**1. Write a class named Employee that has the following member variables:**

**• name. A string that holds the employee’s name.**

**• idNumber. An int variable that holds the employee’s ID number.**

**• department. A string that holds the name of the department where the employee**

**works.**

**• position. A string that holds the employee’s job title.**

**The class should have the following constructors:**

**• A constructor that accepts the following values as arguments and assigns them to**

**the appropriate member variables: employee’s name, employee’s ID number,**

**department, and position.**

**• A constructor that accepts the following values as arguments and assigns them to**

**the appropriate member variables: employee’s name and ID number. The**

**department and position fields should be assigned an empty string ( "" ).**

**• A default constructor that assigns empty strings ( "") to the name, department, and position member variables, and 0 to the idNumber member variable.**

**Write appropriate mutator functions that store values in these member variables and accessor functions that return the values in these member variables. Once you have written the class, write a separate program that creates three Employee objects to hold the following data.**

**PROGRAM**:

#include <iostream>

using namespace std;

class employee {

 private:

  string name;

  int idNumber;

  string department;

  string postion;

 public:

  employee(string *n*, int *i*) {

    name = *n*;

    idNumber = *i*;

    department = "";

    postion = "";

  }

  employee(string *n*, int *i*, string *d*, string *p*) {

    name = *n*;

    idNumber = *i*;

    department = *d*;

    postion = *p*;

  }

  employee() {

    name = "";

    idNumber = 0;

    department = "";

    postion = "";

  }

  void setname(string *n*) { name = *n*; }

  string getname() { return name; }

  void setidNumber(int *i*) { idNumber = *i*; }

  int getidNumber() { return idNumber; }

  void setdepartment(string *d*) { department = *d*; }

  string getdepartment() { return department; }

  void setpostion(string *p*) { postion = *p*; }

  string getpostion() { return postion; }

};

int main() {

  employee e1("Susan Meyers", 47899, "Accounting", "Vice President");

  employee e2("Mark Jones", 39119, "IT", "Programmer");

  employee e3("Joy Rogers", 81774, "Manufacturing", "Engineer");

  cout << e1.getname() << " " << e1.getidNumber() << " " << e1.getdepartment()

       << " " << e1.getpostion() << endl;

  cout << e2.getname() << " " << e2.getidNumber() << " " << e2.getdepartment()

       << " " << e2.getpostion() << endl;

  cout << e3.getname() << " " << e3.getidNumber() << " " << e3.getdepartment()

       << " " << e3.getpostion() << endl;

}

**RESULT**:

A black background with white text

AI-generated content may be incorrect.

**QUESTION#2**

**Create a class named 'Programming'. While creating an object of the class, if nothing is passed to it, then the message "I love programming languages" should be printed. If some String is passed to it, then in place of "programming languages" the name of that String variable should be printed. For example, while creating object if we pass "C++", then "I love C++" should be printed.**

**PROGRAM**:

#include <iostream>

using namespace std;

class programming {

 private:

  string word;

 public:

  programming() { cout << "I love programming languages" << endl; }

  programming(string s) {

    word = s;

    cout << "I love " << word << endl;

  }

};

int main() {

  programming p1;

  programming p2("Umais"); }

**RESULT**:

A black background with white text

AI-generated content may be incorrect.

**QUESTION#3**

**PROGRAM**:

#include <iostream>

using namespace std;

class bank\_account {

 private:

  string name;

  int num;

  int balance;

  int enter;

 public:

  bank\_account() {

    name = "";

    num = 0;

    balance = 0;

  }

  bank\_account(string *n*, int *nu*, int *b*) {

    name = *n*;

    num = *nu*;

    balance = *b*;

  }

  void details() {

    cout << "account number :" << num << endl;

    cout << "account name :" << name << endl;

    cout << " account balace :" << balance << endl;

  }

  void deposit() {

    int dep;

    cout << "Enter amount to deposit :";

    cin >> dep;

    balance += dep;

    cout<<"amount after deposit is :"<<balance<<endl;

  }

  void Withdraw() {

    cout << "Enter amount to withdraw :";

    cin >> enter;

    while (enter > 25000) {

      cout << "Invalid amount ,Enter again:";

      cin >> enter;

    }

    balance -= enter;

    cout<<"amount after withdraw is :"<<balance<<endl;

  }

  void Menu() {

    int opt;

    do {

      cout << "1. deposit amount " << endl;

      cout << "2. withdraw amount " << endl;

      cout << "3. display details " << endl;

      cout << "4. Exit" << endl;

      cout << "Enter option :";

      cin >> opt;

      switch (opt) {

        case 1: {

          deposit();

          cout << "amount deposit sucessfully" << endl;

          break;

        }

        case 2: {

          Withdraw();

          cout << "amount withdraw sucessfully" << endl;

          break;

        }

        case 3: {

          details();

          break;

        }

        default: {

          cout << "invalid option" << endl;

        }

      }

    } while (opt != 4);

  }

};

int main() {

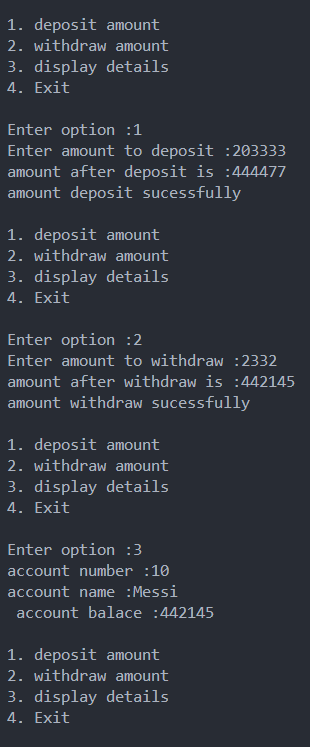
  bank\_account b1;

  bank\_account b2("Messi", 10, 241144);

  b2.Menu();

}

**RESULT**

****

**QUESTION#4:**

**Define a class to represent an item in an online shopping cart. Include the following**

**members:**

**Data Members:**

* **Item ID**
* **Item Name**
* **Price**
* **Quantity in Stock**

**Member Functions:**

* **Create Default and Parameterized Constructors to initialize the item ID, name, price,**
* **and quantity in stock.**
* **Copy Constructor to create a new item object by copying details from an existing**
* **item object.**
* **To add an item to the cart (reduce the quantity in stock by 1 if the item is available).**
* **To remove an item from the cart (increase the quantity in stock by 1).**
* **To display all details of the item.**
* **To display a Menu.**

**Take input from the user for performing the above operations. If the user tries to add an item that is out of stock, display a message indicating that the item is unavailable. Ensure that the user can only add items that are in stock.**

**PROGRAM:**

#include<iostream>

using namespace std;

class shopping{

 private:

 int id;

 string name;

 float price;

 int quantity;

 public:

 shopping(){

    id=0;

   name="";

   price=0.0;

   quantity=0;

}

shopping(int id,string name,float price,int quantity){

      this->id=id;

      this->name=name;

      this->price=price;

      this->quantity=quantity;

}

shopping(shopping &s2){

      this->id=id;

      this->name=name;

      this->price=price;

      this->quantity=quantity;

}

void add\_item(){

    int stock; string check;

    cout<<"Enter item name :";

    cin>>check;

    if(check==this->name){

        cout<<"Enter stock you want to add :";

        cin>>stock;

        quantity+=stock;

        cout<<"new quantity is :"<<quantity<<endl;

    }

    else{

        cout<<"invalid item"<<endl;

    }

}

void remove\_item(){

    int stock; string check;

    cout<<"Enter item name :";

    cin>>check;

    if(check==this->name){

        cout<<"Enter stock you want to remove :";

        cin>>stock;

        while(stock > quantity){

            cout<<"invalid amount,Enter again :";

            cin>>stock;

        }

        quantity-=stock;

        cout<<"new quantity is :"<<quantity<<endl;

    }

    else{

        cout<<"invalid item"<<endl;

    }

}

void displayItem() {

    cout << "Item ID: " << id << endl;

    cout << "Item Name: " << name  << endl;

    cout << "Price: " <<  price << endl;

    cout << "Quantity in Stock: " <<quantity<< endl;

}

void displayMenu() {

    int choice;

    do {

        cout << "\nMENU:\n";

        cout << "1. Add more stock to item" << endl;

        cout << "2. Remove stock from item" << endl;

        cout << "3. Display Item" << endl;

        cout << "4. Exit" << endl;

        cout << "Enter your choice: ";

        cin >> choice;

        switch (choice) {

            case 1:

            add\_item();

                break;

            case 2:

            remove\_item();

                break;

            case 3:

            displayItem();

                break;

            case 4:

                cout << "Exiting program..." << endl;

                break;

            default:

                cout << "Invalid choice. Please try again." << endl;

        }

    } while (choice != 4);

}

};

int main(){

    int id, quantity,opt;

    float price;

    string name;

    shopping s1;

    cout << "Enter Item ID: ";

    cin >> id;

    cin.ignore();

    cout << "Enter Item Name: ";

    getline(cin, name);

    cout << "Enter Price: ";

    cin >> price;

    cout << "Enter Quantity in Stock: ";

    cin >> quantity;

    shopping s2(id, name, price, quantity);

    shopping s3(s2);

    s2.displayMenu();

}

**RESULT**:

**A screenshot of a computer program

AI-generated content may be incorrect.**

**QUESTION#5:**

**Design a class Numbers that can be used to translate whole dollar amounts in the range 0 through 9999 into an English description of the number. For example, the number 713 would be translated into the string seven hundred thirteen, and 8203 would be translated into eight thousand two hundred three. The class should have a single integer member variable: int number; and a static array of string objects that specify how to translate key dollar amounts into the desired format. For example, you might use static strings such as string lessThan20[20] = {"zero", "one", ..., "eighteen", "nineteen"}; string hundred ="hundred"; string thousand = "thousand"; The class should have a constructor that accepts a nonnegative integer and uses it to initialize the Numbers object. It should have a**

**member function print() that prints the English description of the Numbers object. Demonstrate the class by writing a main program that asks the user to enter a number in the proper range and then prints out its English description.**

**PROGRAM**:

#include <iostream>

using namespace std;

class convert {

private:

    int number;

    string lessthan[20] = {"zero", "one", "two", "three", "four", "five", "six", "seven", "eight",

                            "nine","ten","eleven","twelve","thirteen", "fourteen", "fifteen",

                            "sixteen", "seventeen", "eighteen", "nineteen" };

    string thousand = "thousand";

    string hundred = "hundred";

    string tens[10] = {"", "", "twenty", "thirty", "forty", "fifty", "sixty", "seventy", "eighty", "ninety"};

public:

    convert(int *num*) {

        this->number = *num*;

    }

    void print() {

        int length = 0;

        int temp = number;

        while (temp > 0) {

            temp = temp / 10;

            length += 1;

        }

        if (number == 0) {

            cout << "zero" << endl;

            return;

        }

        if (length == 4) {

            if ((number / 1000) % 10 != 0)

                cout << lessthan[(number / 1000) % 10] << " " << thousand << " ";

            number %= 1000;

        }

        if (length >= 3) {

            if ((number / 100) % 10 != 0)

                cout << lessthan[(number / 100) % 10] << " " << hundred << " ";

            number %= 100;

        }

        if (number < 20) {

            if (number != 0)

                cout << lessthan[number];

        } else {

            cout << tens[number / 10];

            if (number % 10 != 0)

                cout << " " << lessthan[number % 10];

        }

    }

};

int main() {

    int number;

    cout << "Enter number between 0-9999: ";

    cin >> number;

    while (number < 0 || number > 9999) {

        cout << "Invalid number, Enter again: ";

        cin >> number;

    }

    convert c1(number);

    c1.print();

    return 0;

}

**RESULT**:

A black background with white text

AI-generated content may be incorrect.