Employee management system:

#include<iostream>

#include<string>

using namespace std;

//Question 1

class employee{

protected :

  string name;

  int id;

public:

  employee(string n = " " , int i = 000){

     name = n;

     id = i;

  }

  virtual void display (){

    cout<<"Employee information \n";

    cout<<"Employee name : "<< name<<endl;

    cout<<"ID : "<<id<<endl;

  }

};

class manager : public employee{

    private:

    string department;

    public:

    manager(string n = " ", int id = 000 , string d = " "): employee(n,id){

        department = d;

    }

    virtual void display(){

        cout<<"Employee name : "<< name<<endl;

        cout<<"ID : "<<id<<endl;

        cout<<"Department: "<< department <<endl;

    }

};

class developer : public employee{

    private:

    string language;

    public:

    developer(string n = " ", int id = 000 , string l = " "): employee(n,id){

        language = l;

    }

    virtual void display(){

        cout<<"Employee name : "<< name<<endl;

        cout<<"ID : "<<id<<endl;

        cout<<"Department: "<< language <<endl;

    }

};

//Question 2 Shape hierarchy :

class shape{

   protected:

   string color;

   public:

   shape(string c = "black"){

     color = c;

   }

   virtual void Area() = 0;

    virtual void display(){

        cout<<"color : "<<color<<endl;

    }

};

class circle : public shape{

    private :

    double radius;

    double area;

    public:

    circle( double r = 0.0 ,string c = "black" ): shape(c){

        radius= r;

    }

    virtual void Area(){

        area = radius \* radius;

    }

    virtual void display(){

        cout<<"color : "<<color<<endl;

        cout<<"Area of circle : "<<area<<endl;

    }

};

class rectangle : public shape{

    private :

    double length , width;

    double area;

    public:

    rectangle( double l = 0.0 ,double w = 0.0 ,string c = "black" ): shape(c){

        length = l;

        width = w;

    }

    virtual void Area(){

        area = length \* width;

    }

    virtual void display(){

        cout<<"color : "<<color<<endl;

        cout<<"Area of rectangle : "<<area<<endl;

    }

};

// QUESTION 3 animal kingdom:

class animalkingdom{

protected:

string name;

public:

animalkingdom(string n = "x"){

    name = n;

}

virtual void display(){

    cout<<"Name : "<< name <<endl;

}

};

class dog : public animalkingdom{

private :

string type;

public:

dog(string n = "x" , string t = "simple"): animalkingdom(n){

    type = t;

}

virtual void display(){

    cout<<"This is a dog \n";

    cout<<"Name : "<< name <<endl;

    cout<<"Type : "<< type <<endl;

}

};

class cat : public animalkingdom{

private :

string type;

public:

cat(string n = "x" , string t = "simple"): animalkingdom(n){

    type = t;

}

virtual void display(){

    cout<<"This is a cat \n";

    cout<<"Name : "<< name <<endl;

    cout<<"Type : "<< type <<endl;

}

};

class bird : public animalkingdom{

private :

string type;

public:

bird(string n = "x" , string t = "simple"): animalkingdom(n){

    type = t;

}

virtual void display(){

    cout<<"This is a bird \n";

    cout<<"Name : "<< name <<endl;

    cout<<"Type : "<< type <<endl;

}

};

//QUESTION 4 bank account system:

class account {

protected :

    string accountid;

    string accountype;

    int balance;

public:

    account(string id = "xyz123", string type = "savings", int b = 100 ) {

        accountid = id;

        accountype = type;

        balance = b;

    }

    void withdraw(int b) {

        balance = balance - b;

    }

     virtual void display() {

        cout << "Account ID " << accountid << endl;

        cout << "Account Type " << accountype << endl;

        cout << "Account balance  " << balance << endl;

    }

};

class savingsaccount : public account {

private:

    int months;

public:

    savingsaccount(string id = "xyz123", string type = "savings", int b = 100,int m = 0) : account(id, type, b) {

        months = m;

    }

    virtual void display() {

        cout << "Account ID " << accountid << endl;

        cout << "Account Type " << accountype << endl;

        cout << "Account balance  " << balance << endl;

        cout << "months passed " << months<<endl<<endl;

    }

};

class checkingaccount : public account {

private:

    int overdraftlimit;

public:

    checkingaccount(string id = "xyz123", string type = "savings", int b = 100, int limit = 5000) : account(id, type, b) {

        overdraftlimit = limit;

    }

    virtual void display() {

        cout << "Account ID " << accountid << endl;

        cout << "Account Type " << accountype << endl;

        cout << "Account balance  " << balance << endl;

        cout << "overdraft limit  " << overdraftlimit<<endl<<endl;

    }

};

//Question 5 vehicle rental system:

class vehicle{

protected:

int year;

public:

vehicle(int n ){

    year = n;

}

virtual void display(){

    cout<<"Purchase Year of vehicle : "<< year<<endl;

}

};

class car : public vehicle{

private :

int engine;

string name;

public:

car (int y = 2015 , int e = 660 , string n = "alto"): vehicle(y){

    engine = e;

    name = n;

}

virtual void display(){

    cout<<"Engine "<< engine << endl;

    cout<<"Name "<< name<<endl;

    cout<<"Purchase Year of vehicle : "<< year<<endl<<endl;

}

};

class motorcycle : public vehicle{

private :

string name;

public:

motorcycle (int y = 2015 , string n = "yamaha"): vehicle(y){

    name = n;

}

virtual void display(){

    cout<<"Name "<< name<<endl;

    cout<<"Purchase Year of vehicle : "<< year<<endl<<endl;

}

};

//QUESTION 6 student management system:

class student  {

protected:

   string name;

    float cgpa;

    int age ;

public:

    student(string n = " " , int a= 18,  float c= 0.0)  {

        name = n;

        age = a;

        cgpa = c;

    }

    virtual void display(){

        cout<<"Name "<< name<<endl;

        cout<<"Cgpa "<< cgpa<<endl;

        cout<<"Age "<< age<<endl<<endl;

    }

};

class undergraduate : public student {

private:

    string fypname;

public:

    undergraduate(string n = " ",  int age = 18,  float c = 0.0 , string fyp = " "): student(n,age,c) {

    fypname = fyp;

        cout << "Undergraduate called \n";

    }

    virtual void display(){

        cout<<"Name "<< name<<endl;

        cout<<"Cgpa "<< cgpa<<endl;

        cout<<"Age "<< age<<endl;

        cout<<"project "<<fypname<<endl<<endl;

        }

};

class graduate : public student {

private:

    string thesis;

public:

    graduate(string n = " ",  int age = 18,  float c = 0.0, string t = " ") : student(n, age, c) {

        thesis = t;

    }

    virtual void display(){

        cout<<"Name "<< name<<endl;

        cout<<"Cgpa "<< cgpa<<endl;

        cout<<"Age "<< age<<endl;

        cout<<"thesis "<<thesis<<endl<<endl;

        }

};

//Question 7 library management system:

class Item {

protected:

    string title;

    string authorOrDirector;

    int year;

public:

    Item(string t, string aod, int y) : title(t), authorOrDirector(aod), year(y) {}

    virtual void displayInfo() const {

        cout << "Title: " << title << ", Author/Director: " << authorOrDirector << ", Year: " << year;

    }

};

class Book : public Item {

private:

    string genre;

public:

    Book(string t, string aod, int y, string g) : Item(t, aod, y), genre(g) {}

    virtual void displayInfo() {

       cout << "Title: " << title << ", Author/Director: " << authorOrDirector << ", Year: " << year;

        cout << ", Genre: " << genre << endl;

    }

};

class DVD : public Item {

private:

    string genre;

public:

    DVD(string t, string aod, int y, string g) : Item(t, aod, y), genre(g) {}

    virtual void displayInfo() {

        cout << "Title: " << title << ", Author/Director: " << authorOrDirector << ", Year: " << year;

        cout << ", Genre: " << genre << endl;

    }

};

//Question 8 employee payroll system:

class Employee {

    private:

    int id\_;

    string name\_;

public:

    Employee(int id, string name) : id\_(id), name\_(name) {}

    virtual double calculate\_pay() const = 0;

    virtual void display\_info() const {

        cout << "Employee ID: " << id\_ << endl;

        cout << "Name: " << name\_ << endl;

    }

};

class HEmployee : public Employee {

    private:

    double wage\_;

    double hours\_worked\_;

public:

    HEmployee(int id, string name, double wage, double hours\_worked) :

        Employee(id, name), wage\_(wage), hours\_worked\_(hours\_worked) {}

    double calculate\_pay() const override {

        return wage\_ \* hours\_worked\_;

    }

    void display\_info() const override {

        Employee::display\_info();

        cout << "Wage: $" << wage\_ << endl;

        cout << "Hours Worked: " << hours\_worked\_ << endl;

    }

};

class SEmployee : public Employee {

    private:

    double salary\_;

public:

    SEmployee(int id, string name, double salary) :

        Employee(id, name), salary\_(salary) {}

    double calculate\_pay() const override {

        return salary\_;

    }

    void display\_info() const override {

        Employee::display\_info();

        cout << "Salary: $" << salary\_ << endl;

    }

};

// Question 9 product inventory system:

class Product {

private:

    int productid;

    string name\_;

    double price;

    int quantity;

public:

    Product(int product\_id, string name, double price, int quantity) :

        productid(product\_id), name\_(name), price(price), quantity(quantity) {}

    virtual void display\_info() const {

        cout << "Product ID: " << productid << endl;

        cout << "Name: " << name\_ << endl;

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

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

    }

};

class Electronics : public Product {

private:

    string brand;

public:

    Electronics(int product\_id, string name, double price, int quantity, string brand) :

        Product(product\_id, name, price, quantity), brand(brand) {}

    void display\_info() const override {

        Product::display\_info();

        cout << "Brand: " << brand << endl;

    }

};

class Clothing : public Product {

private:

    string size\_;

public:

    Clothing(int product\_id, string name, double price, int quantity, string size) :

        Product(product\_id, name, price, quantity), size\_(size) {}

    void display\_info() const override {

        Product::display\_info();

        cout << "Size: " << size\_ << endl;

    }

};

// QUESTION 10 restaurant menu system:

class MenuItem

{

protected:

    string name;

    double price;

public:

    MenuItem(const string& n, double p) : name(n), price(p) {}

    virtual void display() const

    {

        cout << "Name: " << name << ", Price: $" << price << endl;

    }

};

class Appetizer : public MenuItem

{

public:

    Appetizer(const string& n, double p) : MenuItem(n, p) {}

    void display() const override

    {

        cout << "Appetizer - ";

        MenuItem::display();

    }

};

class MainCourse : public MenuItem

{

public:

    MainCourse(const string& n, double p) : MenuItem(n, p) {}

    void display() const override

    {

        cout << "Main Course - ";

        MenuItem::display();

    }

};

class Dessert : public MenuItem

{

public:

    Dessert(const string& n, double p) : MenuItem(n, p) {}

    void display() const override

    {

        cout << "Dessert - ";

        MenuItem::display();

    }

};

int main(){

    int num ;

    cout<<"enter which question : "; cin >> num;

    if (num == 1){

        const int n = 2;

        employee \* employees [n];

        employees[0] = new manager ("zainab ", 001 , "cybersecuirty");

        employees[1] = new developer ("fatima ", 002 , "python");

        employees[2] = new developer ("ali ", 003, "c++");

        for(int i = 0 ; i <= n ; i++){

            employees[i]->display();

        }

        delete[] employees;

    }

    else if (num == 2){

        const int n = 2 ;

        shape \* shapes [n];

        shapes[0] = new circle(2.0, "red");

        shapes[1] = new circle(2.0, "blue");

        shapes[2] = new rectangle(2.0, 3.0 , "pruple");

        for(int i = 0 ; i <= n ; i++){

            shapes[i]->Area();

            shapes[i]->display();

        }

        delete[] shapes;

    }

    else if(num == 3){

       const int n = 2 ;

        animalkingdom \* animals [n];

        animals[0] = new dog("spike" , "husky");

        animals[1] = new cat("puffy", "persian cat");

        animals[2] = new bird("tommy" , "peacock");

        for(int i = 0 ; i <= n ; i++){

            animals[i]->display();

        }

        delete[] animals;

    }

    else if (num == 4){

        const int n = 2 ;

        account \* accounts [n];

        accounts[0] = new savingsaccount("A123", "savings",10000, 20);

        accounts[1] = new savingsaccount("A456", "savings",13000, 10);

        accounts[2] = new checkingaccount("B123", "savings",10000);

        for(int i = 0 ; i <= n ; i++){

            accounts[i]->display();

        }

        delete[] accounts;

    }

    else if (num == 5){

        const int n = 2 ;

        vehicle \* vehicles [n];

        vehicles[0] = new car(2020 , 1500, "BRV");

        vehicles[1] = new motorcycle(2023, "YZF-R1");

        vehicles[2] = new motorcycle(2024, "YZF-R6");

        for(int i = 0 ; i <= n ; i++){

            vehicles[i]->display();

        }

        delete[] vehicles;

    }

    else if (num == 6){

        const int n = 2 ;

        student \* students [n];

        students[0] = new undergraduate("zainab", 18 , 3.4 , "candycrush");

        students[1] = new graduate("fatima " , 24 , 3.0 , "oop");

        students[2] = new undergraduate("ali", 20 , 3.6 , "management");

        for(int i = 0 ; i <= n ; i++){

            students[i]->display();

        }

        delete[] students;

    }

    else if (num == 7 ){

        const int n = 2 ;

        Item \* items [n];

        items[0] = new Book("alchemist", "paulo coelho" , 2010 , "philosophy");

        items[1] = new DVD("MI","abcxyz", 2019 , "action");

        items[2] = new Book("famoufive", "enid" , 2000 , "fiction");

        for(int i = 0 ; i <= n ; i++){

            items[i]->displayInfo();

        }

        delete [] items;

    }

    else if (num == 8 ){

    Employee\* employee1 = new HEmployee(573, "zainab", 500000, 40.0);

    Employee\* employee2 = new SEmployee(516, "ali", 10.00);

    Employee\* employees[] = {employee1, employee2 };

    for (int i = 0; i < sizeof(employees) / sizeof(employees[0]); i++) {

        employees[i]->display\_info();

        cout << "Pay: $" << employees[i]->calculate\_pay() << endl << endl;

    }

    delete employee1;

    delete employee2;

    }

    else if ( num == 9 )

    {

    Product\* product1 = new Electronics(100, "Laptop", 599.99, 20, "Dell");

    Product\* product2 = new Clothing(200, "Shirt", 19.99, 50, "small");

    Product\* inventory[] = { product1, product2 };

    for (int i = 0; i < sizeof(inventory) / sizeof(inventory[0]); i++) {

        inventory[i]->display\_info();

        cout << endl;

    }

    delete product1;

    delete product2;

    }

    else if ( num == 10){

    MenuItem\*\* menu = new MenuItem \* [3];

    menu[0] = new Appetizer("Salad", 5.22);

    menu[1] = new MainCourse("pizza", 10.99);

    menu[2] = new Dessert("Cake", 6.99);

    cout << "Restaurant Menu:" << endl;

    for (int i = 0; i < 3; ++i)

    {

        menu[i]->display();

    }

    for (int i = 0; i < 3; ++i)

    {

        delete menu[i];

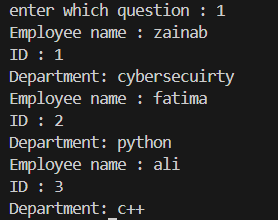
    }

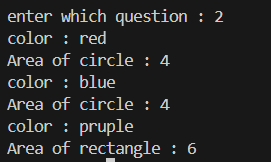
    delete[] menu;

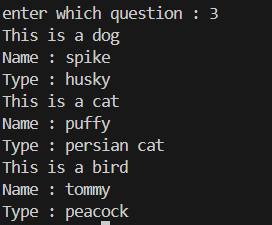
    }

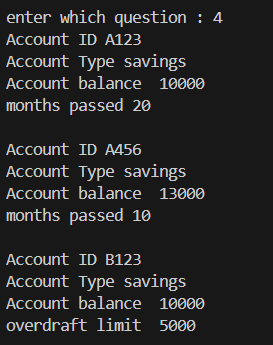
}

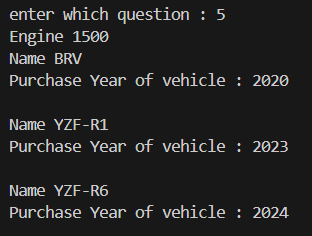
**OUTPUTS:**

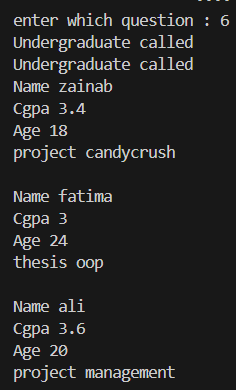
****

****

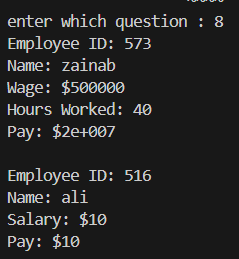
****

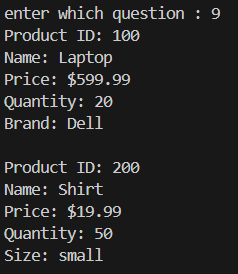
****

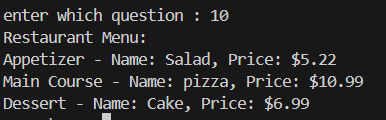
****

****

****

****

****

****