

**ASSIGNMENT NO: 02**

**SUBMITTED BY : Hafiza Tehreem fatima**

**SUBMITTED TO: SIR JAVED**

**Registration no: 026**

**DEPARTEMENT : BS-AI**

**Problem 1:**

//file: class.cpp

//date: 01-05-2024

//name: tehreem fatima

//registration no: 2023-bs-ai-026

//question statement : Write a user-defined program to declare a class which stores a complex number. Demonstrate the use of constant objects, constant member function and constant arguments, using this class.

#include <iostream>

using namespace std;

**Output:**

3 + 4i

2 + 3i

10 + 3i

class Complex {

private:

double real, imag;

public:

Complex(double r = 0, double i = 0) : real(r), imag(i) {}

void display() const {

cout << real << " + " << imag << "i" <<endl;

}

void setReal(double r) {

real = r;

}

};

int main() {

const Complex c1(3.0, 4.0);

c1.display(); // 3+ 4i

Complex c2(2.0, 3.0);

c2.display(); // 2 + 3i

c2.setReal(10);

c2.display(); // 10 + 3i

}

**Problem 2**

//file: car.cpp

//date:01-05-2024

//name: tehreem fatima

//registration no: 2023-bs-ai-026

//question statement: Write a class that contain the following attribute

. The name of car

.Direction of car (E,W,N,S)

. The position of car ( from imaginary to zero point )

The class has fallowing member function

The constructor to be initialize

• Turn function use to change the direction of car to one steps right side (e.g. if the direction is E, Should be change to S and so on)

• Overload the turn function to change the direction to any side directly. It should accept the direction parameter.

• Move function to change the position of car away from zero point. It should accept the distance as parameter.

#include <iostream>

using namespace std;

class Car {

private:

string name;

char direction;

int position;

public:

Car(string n) : name(n), direction('N'), position(0) {}

void turn() {

direction = (direction == 'N') ? 'E' : (direction == 'E') ? 'S' : (direction == 'S') ? 'W' : 'N';

}

void turn(char dir) {

**Output:**

Car: civic, Direction: N, Position: 0

Car: civic, Direction: E, Position: 0

Car: civic, Direction: S, Position: 0

Car: civic, Direction: S, Position: 10

direction = dir;

}

void move(int dist) {

position += dist;

}

void display() const {

cout << "Car: " << name << ", Direction: " << direction << ", Position: " << position << endl;

}

};

int main() {

Car myCar("civic");

myCar.display();

myCar.turn();

myCar.display();

myCar.turn('S');

myCar.display();

myCar.move(10);

myCar.display();

return 0;

}

**Problem 3**

//file: dimention array.cpp

//date: 01-05-2024

//name: tehreem fatima

//registration no: 2023-bs-ai-026

//question statement: Write a function find(...) that accepts a one-dimensional integer array of size 10 as an argument to the function. Your program then finds the location and value of the largest and second-largest elements in a one-dimensional array. Display answers in main().

#include <iostream>

using namespace std;

void find(int arr[10]) {

int max = arr[0];

int secondMax = arr[0];

int maxIndex = 0;

int secondMaxIndex = 0;

**Output:**

Largest element: 10

Second largest element: 9

for (int i = 1; i < 10; i++) {

if (arr[i] > max) {

secondMax = max;

max = arr[i];

} else if (arr[i] > secondMax) {

secondMax = arr[i];

}

}

cout << "Largest element: " << max <<endl;

cout << "Second largest element: " << secondMax <<endl;

}

int main() {

int arr[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

find(arr);

}

**Problem 4**

//file:one dimentional integar.cpp

//date:01-05-2024

//name:tehreem fatima

//registration no:2023-bs-ai-026

//question statement: Write a function arrange...) that accepts a one-dimensional integer array of size 10 as an argument to the function. The program then shifts negative numbers to the left and positive numbers to the right side of the array.

#include <iostream>

using namespace std;

void arrange(int arr[10]) {

int j = 0; // index for negative numbers

for (int i = 0; i < 10; i++) {

if (arr[i] < 0) {

int temp = arr[j];

arr[j] = arr[i];

arr[i] = temp;

j++;

}

}

**Output:**

-2 -3 -7 4 5 6 1 8 9 10

}

int main() {

int arr[10] = {1, -2, -3, 4, 5, 6, -7, 8, 9, 10};

arrange(arr);

for (int i = 0; i < 10; i++) {

cout << arr[i] << " ";

}

}

**Problem 5**

//file: class employe.cpp

//date: 01-05-2024

//name: tehreem fatima

//registration no: 2023-bs-ai-026

//question statement: Create a class employee which stores is name, ID and salary of an employee by user input. The ID should be generated upon the creation of object, starting from 1. Include all the constructors and destructor in the class. Create one object using each of the constructors and display it.

#include <iostream>

#include <string>

using namespace std;

class Employee {

private:

string name;

int id;

double salary;

static int nextID;

public:

Employee() : id(nextID++) {}

Employee(string n, double s) : name(n), id(nextID++), salary(s) {}

void display() {

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

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

cout << "Salary: " << salary <<endl;

}

};

Name:

ID: 1

Salary: 2.56838e+261

Name: sunny

ID: 2

Salary: 20000

Name: sunny

ID: 2

Salary: 20000

int Employee::nextID = 1;

int main() {

Employee e1, e2("sunny", 20000.0), e3(e2);

e1.display();

e2.display();

e3.display();

return 0;

}

**Problem 6**

//file: transport vehicle.cpp

//date: 01-05-2024

//name: tehreem fatima

//registration no: 2023-bs-ai-026

//question statement: Write a C++ program for the class vehicle and its drive class water transport, road transport and air transport vehicles. Make suitable data variables and member functions. When you create an object must be count and display total no of object created also create every class objects and access member through the member functions.

#include <iostream>

using namespace std;

class Vehicle {

protected:

static int count;

string name;

public:

Vehicle(string n) : name(n) {

count++;

}

virtual void display() {

cout << "Vehicle: " << name <<endl;

}

**Output:**

Vehicle: Vehicle1

Vehicle: WaterTransport1

Water Transport

Vehicle: RoadTransport1

Road Transport

Vehicle: AirTransport1

Air Transport

Total Vehicles: 4

static int getTotal() {

return count;

}

};

int Vehicle::count = 0;

class WaterTransport : public Vehicle {

public:

WaterTransport(string n) : Vehicle(n) {}

void display() {

Vehicle::display();

cout << "Water Transport" <<endl;

}

};

class RoadTransport : public Vehicle {

public:

RoadTransport(string n) : Vehicle(n) {}

void display() {

Vehicle::display();

cout << "Road Transport" <<endl;

}

};

class AirTransport : public Vehicle {

public:

AirTransport(string n) : Vehicle(n) {}

void display() {

Vehicle::display();

cout << "Air Transport" <<endl;

}

};

int main() {

Vehicle v1("Vehicle1");

WaterTransport wt1("WaterTransport1");

RoadTransport rt1("RoadTransport1");

AirTransport at1("AirTransport1");

v1.display();

wt1.display();

rt1.display();

at1.display();

cout << "Total Vehicles: " << Vehicle::getTotal() <<endl;

}

**Problem 7**

//file: employe.cpp

//date: 01-05-2024

//name: tehreem fatima

//registration no: 2023-bs-ai-026

//question statement: Implement a C++ class named Employee with the following specifications:

• The class should have private data members name (string), id (integer), and salary (floating-point).

• Implement a static data member totalEmployees to keep track of the total number of employees.

• Implement a static member function averageSalary) that calculates and returns the average salary of all employees.

• Provide member functions to set and get the values of name, id, and salary.

: Implement a destructor intime the ate i lo as cou when an eie is destroyed.

#include <iostream>

#include <string>

using namespace std;

**Output:**

Name:

ID: 1

Salary: 2.56838e+261

Name: abraham

ID: 2

Salary: 4000

Name: abraham

ID: 2

Salary: 4000

class Employee {

private:

string name;

int id;

double salary;

static int nextID;

public:

Employee() : id(nextID++) {}

Employee(string n, double s) : name(n), id(nextID++), salary(s) {}

void display() {

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

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

cout << "Salary: " << salary <<endl;

}

};

int Employee::nextID = 1;

int main() {

Employee e1, e2("abraham", 4000.0), e3(e2);

e1.display();

e2.display();

e3.display();

}

**Problem 8**

//file: car pooling.cpp

//date:01-05-2024

//name: tehreem fatima

//registration no: 2023-bs-ai-026

//question statement: (Car Pool Savings Calculator) Research several car-pooling websites. create an application that calculates your daily driving cost, so that you can estimate how much money could be saved by carpooling, which also has other advantages such as reducing carbon emission and reducing traffic congestion. The application should input the following and display the user's cost per day of driving to word:

a) Total miles driven per day.

b) Cost per gallon of gasoline.

c) Average miles per gallon.

d) Parking fees per day.

e) Toll per day.

#include <iostream>

using namespace std;

int main() {

double milesDriven, costPerGallon, milesPerGallon, parkingFees, toll;

cout << "Enter total miles driven per day: ";

**Output:**

Enter total miles driven per day: 5

Enter cost per gallon of gasoline: 1000

Enter average miles per gallon: 7

Enter parking fees per day: 20

Enter toll per day: 30

Your daily driving cost is: $764.286

cin >> milesDriven;

cout << "Enter cost per gallon of gasoline: ";

cin >> costPerGallon;

cout << "Enter average miles per gallon: ";

cin >> milesPerGallon;

cout << "Enter parking fees per day: ";

cin >> parkingFees;

cout << "Enter toll per day: ";

cin >> toll;

double gasCost = milesDriven / milesPerGallon \* costPerGallon;

double totalCost = gasCost + parkingFees + toll;

cout << "Your daily driving cost is: $" << totalCost <<endl;

}