TRIBHUVAN UNIVERSITY

**PATAN MULTIPLE CAMPUS**

PATAN DHOKA, LALITPUR

**OBJECT ORIENTED PROGRAMMING (BIT 153)**

**LAB 1**

|  |  |
| --- | --- |
| **SUBMITTED BY** | **SUBMITTED TO** |
|  |  |
| NAME: SURESH DAHAL | DADHI RAM GHIMIRE |
| CLASS: BIT – I/II |  |
| ROLL NO: 23 | ………………………… |
| DATE: 2081/03/30 | CHECKED BY |

**1. Write a C++ Program to print the following using cout and manupulators (endl, left, right, setw)**



## **PROGRAM**

#include <iostream>

#include <iomanip>

using namespace std;

int main() {

cout << left << setw(10) << "Sno."

<< setw(20) << "City"

<< setw(20) << "District"

<< setw(15) << "Province"

<< setw(12) << "Population" << endl;

cout << left << setw(10) << "1."

<< setw(20) << "Kathmandu"

<< setw(20) << "Kathmandu"

<< setw(15) << "Bagmati"

<< setw(12) << "500000" << endl;

cout << left << setw(10) << "2."

<< setw(20) << "Pokhara"

<< setw(20) << "Gandaki"

<< setw(15) << "Kaski"

<< setw(12) << "50000" << endl;

cout << left << setw(10) << "3."

<< setw(20) << "Butwal"

<< setw(20) << "Rupandehi"

<< setw(15) << "Lumbini"

<< setw(12) << "100000" << endl;

cout << left << setw(10) << "4."

<< setw(20) << "Dharan"

<< setw(20) << "Sunsari"

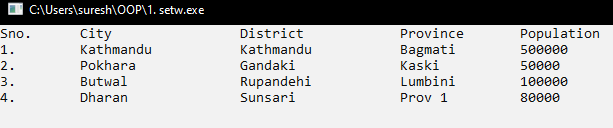
<< setw(15) << "Prov 1"

<< setw(12) << "80000" << endl;

return 0;

}

## **OUTPUT**



**2. Write a Program defining an inline function to compute the area of circle with radius as input.**

**PROGRAM**

#include<iostream>

using namespace std;

inline float calculate\_area(int r) {

return 3.1415 \* r \* r;

}

int main() {

int r;

cout<<"Enter radius of circle: ";

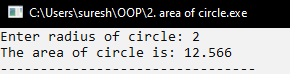
cin>>r;

cout << "The area of circle is: "<<calculate\_area(r);

return 0;

}

**OUTPUT**



**3. Write a program to get input a string and print the string and its reverse. Define your own function to reverse the string in your program.**

**PROGRAM**

#include<iostream>

using namespace std;

void reverse\_string(char str[]) {

int len = 0;

while(str[len] != '\0') {

len++;

}

char rev\_str[len];

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

rev\_str[i] = str[len - i - 1];

}

cout<<"Reversed string is: " << rev\_str;

}

int main() {

char str[30];

cout<<"Enter a string: ";

cin>>str;

reverse\_string(str);

return 0;

}

**OUTPUT**

****

**4. Write a program overloading function sort() to sort an array of integers, characters as well as floating point numbers(Use any sorting algorithm).**

**PROGRAM**

#include<iostream>

using namespace std;

void sort(int arr[]) {

for(int i=0; i<5; i++) {

for(int j=i; j<5; j++) {

if(arr[i] > arr[j]) {

int temp = arr[j];

arr[j] = arr[i];

arr[i] = temp;

}

}

}

cout<<"Sorted array: "<<endl;

for(int i=0; i<5; i++) {

cout<<arr[i]<<endl;

}

}

void sort(float arr[]) {

for(int i=0; i<5; i++) {

for(int j=i; j<5; j++) {

if(arr[i] > arr[j]) {

int temp = arr[j];

arr[j] = arr[i];

arr[i] = temp;

}

}

}

cout<<"Sorted array: "<<endl;

for(int i=0; i<5; i++) {

cout<<arr[i]<<endl;

}

}

int main() {

int arr[5];

float f\_arr[5];

cout<<"Enter integer elements: ";

for(int i=0; i<5; i++) {

cin>>arr[i];

}

cout<<"Enter float elements: ";

for(int i=0; i<5; i++) {

cin>>f\_arr[i];

}

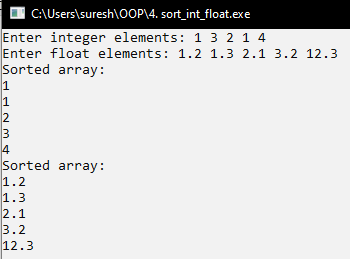
sort(arr);

sort(f\_arr);

return 0;

}

**OUTPUT**

****

**5. Write a program that uses a structure Distance with data members meter and centimeter. Add functions in structure to take input and output as well as the function to add the two variables of Distance and return the sum. Your program should display the result.**

**PROGRAM**

#include <iostream>

using namespace std;

struct Distance {

int meter;

int centimeter;

void inputDistance() {

cout << "Enter meter: ";

cin >> meter;

cout << "Enter centimeter: ";

cin >> centimeter;

}

void outputDistance() {

cout << "Distance: " << meter << " meters and " << centimeter << " centimeters" << endl;

}

Distance addDistance(Distance d2) {

Distance sum;

sum.meter = meter + d2.meter;

sum.centimeter = centimeter + d2.centimeter;

if (sum.centimeter >= 100) {

sum.meter += sum.centimeter / 100;

sum.centimeter = sum.centimeter % 100;

}

return sum;

}

};

int main() {

Distance distance1, distance2, sum;

cout << "Enter details for Distance 1:" << endl;

distance1.inputDistance();

cout << "Enter details for Distance 2:" << endl;

distance2.inputDistance();

sum = distance1.addDistance(distance2);

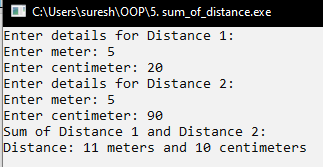
cout << "Sum of Distance 1 and Distance 2:" << endl;

sum.outputDistance();

return 0;

}

**OUTPUT**

****

**6. Write program with objects as function(to add time) argument by passing by value, passing by address and passing by reference defining a class Time with data member hour, minute, second as integers. Write member functions to read the data for objects and to show the value of objects of Time.**

**PROGRAM**

#include <iostream>

using namespace std;

class Time {

private:

int hour;

int minute;

int second;

public:

void readTime() {

cout << "Enter hours: ";

cin >> hour;

cout << "Enter minutes: ";

cin >> minute;

cout << "Enter seconds: ";

cin >> second;

}

void displayTime() {

cout << "Time: " << hour << " hours " << minute << " minutes " << second << " seconds" << endl;

}

Time addByValue(Time t) {

Time sum;

sum.hour = hour + t.hour;

sum.minute = minute + t.minute;

sum.second = second + t.second;

if (sum.second >= 60) {

sum.minute += sum.second / 60;

sum.second = sum.second % 60;

}

if (sum.minute >= 60) {

sum.hour += sum.minute / 60;

sum.minute = sum.minute % 60;

}

return sum;

}

void addByReference(Time &t) {

hour += t.hour;

minute += t.minute;

second += t.second;

if (second >= 60) {

minute += second / 60;

second = second % 60;

}

if (minute >= 60) {

hour += minute / 60;

minute = minute % 60;

}

}

};

int main() {

Time time1, time2;

cout << "Enter details for Time 1:" << endl;

time1.readTime();

cout << "Enter details for Time 2:" << endl;

time2.readTime();

cout << "\nTime 1:" << endl;

time1.displayTime();

cout << "\nTime 2:" << endl;

time2.displayTime();

Time sumByValue = time1.addByValue(time2);

cout << "\nSum of Time 1 and Time 2 (using pass by value):" << endl;

sumByValue.displayTime();

time1.addByReference(time2);

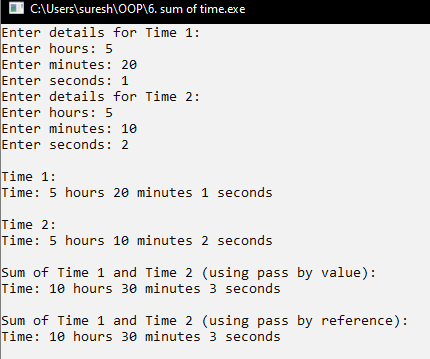
cout << "\nSum of Time 1 and Time 2 (using pass by reference):" << endl;

time1.displayTime();

return 0;

}

**OUTPUT**

****