**Prof. Ramkrishna More**

**Arts, Commerce and Science College(Autonomous)**

**Akurdi Pradhikaran, Pune-44**

**M. SC - COMPUTER APPLICATIONS**



Semester-I

Lab Course: C++

(CAMAP-)

WorkBook

**Name of Student: Ms. Jagruti Krushna Shirsath**

**Roll No: 35**

**Academic Year: 2024-25**

|  |  |
| --- | --- |
| **Program Number** | **Question** |
| 1 | Write a program in C++ to prompt the user to input her/his name and print this name on the screen. |
| 2 | Write a program in C++ to display various types of arithmetic operations (Menu driven). |
| 3 | Write a C++ program to prompt the user to input 3 integer values and print these values in forward and reversed order. |
| 4 | Write a program to find the greatest of three numbers. |
| 5 | Write a program in C++ to swap two numbers with and without using a temporary variable. |
| 6 | Write a C++ program to calculate the area of a circle, rectangle, and triangle using function overloading. |
| 7 | Write a C++ program to sort an array of numbers in descending order. |
| 8 | Write a C++ program to accept and display employee details using the this pointer. |
| 9 | Write a C++ program to print the area of a circle, square, and rectangle using an inline function. |
| 10 | Write a C++ program to read an integer 'n' and print the factorial of 'n' using a recursive function. |
| 11 | Write a C++ program to compute the sum of a specified number of prime numbers. |
| 12 | Write a program in C++ to demonstrate the manipulators: endl, setw, and setfill. |
| 13 | Write a C++ program to demonstrate the use of a Friend function in a class. |
| 14 | Write a C++ program to sort an array in ascending order using selection sort. |
| 15 | Write a C++ program to create a class Person to accept and display five persons' details. |
| 16 | Write a C++ program to create a class Student with attributes and display percentage and class obtained. |
| 17 | Create a class for a student object with attributes like roll number, name, number of subjects, and marks. |
| 18 | Write a class Complex with constructors and overload the + and - operators. |
| 19 | Write a C++ program to find the volume of a cube, cylinder, and rectangular prism using function overloading. |
| 20 | Write a C++ program to find the area of a triangle, circle, and rectangle using function overloading. |
| 21 | Create a class Student to accept and display details for students. |
| 22 | Create two classes dist1 and dist2 to accept distances and find the sum and difference using a friend function. |
| 23 | Write a C++ program using classes with integer data members, object initialization, and a function to display the maximum value. |
| 24 | Write a C++ program to create a class Person that contains data members as Person\_Name, City, Mob\_No. Write a C++ program to perform following functions: a. To accept and display Person information b. To search the mobile number of a given person c. To search the Person details of a given mobile number (Use Function Overloading) |
| 25 | Create a class Time with hours, minutes, and seconds as data members; overload the + operator to add two time objects. |

**Certificate**

This is to certify that Mr./Ms. has successfully completed the programming assignments in the Advanced Database Laboratory.

(Dr. Santosh Jagtap)

**Lab In-charge HOD**

**Internal Examiner External Examiner**

**1.Write a C++ program to prompt the user to input her/his name and print this name on the screen, as shown below. The text from keyboard can be read by using cin>> and to display the text on the screen you can use cout<<.**

**Code:**

#include <iostream>

#include <string>

using namespace std;

int main() {

string name;

cout << "Please enter your name: ";

getline(cin, name);

cout << "Hello, " << name << "!" << endl;

return 0;

}

**Output:**

**Please enter your name: Jagruti Shirsath**

**Hello, Jagruti Shirsath!**

**2. Write a program in C++ to display various type or arithmetic operation (Menu driven)**

**Code:**

#include <iostream>

using namespace std;

void addition(double a, double b);

void subtraction(double a, double b);

void multiplication(double a, double b);

void division(double a, double b);

int main() {

int choice;

double num1, num2;

do {

cout << "Arithmetic Operations Menu" << endl;

cout << "1. Addition" << endl;

cout << "2. Subtraction" << endl;

cout << "3. Multiplication" << endl;

cout << "4. Division" << endl;

cout << "5. Exit" << endl;

cout << "Enter your choice (1-5): ";

cin >> choice;

if (choice == 5) {

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

break;

}

cout << "Enter first num: ";

cin >> num1;

cout << "Enter second num: ";

cin >> num2;

switch (choice) {

case 1:

addition(num1, num2);

break;

case 2:

subtraction(num1, num2);

break;

case 3:

multiplication(num1, num2);

break;

case 4:

division(num1, num2);

break;

default:

cout << "Invalid choice! Please choose between 1 and 5." << endl;

}

cout << endl;

} while (choice != 5);

return 0;

}

void addition(double a, double b) {

cout << "Result: " << a + b << endl;

}

void subtraction(double a, double b) {

cout << "Result: " << a - b << endl;

}

void multiplication(double a, double b) {

cout << "Result: " << a \* b << endl;

}

void division(double a, double b) {

if (b != 0) {

cout << "Result: " << a / b << endl;

}

else {

cout << "Zero division error." << endl;

}

}

**Output:**

Arithmetic Operations Menu

1. Addition

2. Subtraction

3. Multiplication

4. Division

5. Exit

Enter your choice (1-5): 3

Enter first num: 20

Enter second num: 5

Result: 100

Arithmetic Operations Menu

1. Addition

2. Subtraction

3. Multiplication

4. Division

5. Exit

Enter your choice (1-5):

**3.Write a C++ program to prompt the user to input 3 integer values and print these values in forward and reversed order.**

**Code:**

#include <iostream>

using namespace std;

int main() {

int num1, num2, num3;

cout << "Enter three integer values:" << endl;

cout << "Enter the first num: ";

cin >> num1;

cout << "Enter the second num: ";

cin >> num2;

cout << "Enter the third num: ";

cin >> num3;

cout << "Values in forward order: " << num1 << ", " << num2 << ", " << num3 << endl;

cout << "Values in reversed order: " << num3 << ", " << num2 << ", " << num1 << endl;

return 0;

}

**Output:**

Enter three integer values:

Enter the first num: 12

Enter the second num: 30

Enter the third num: 10

Values in forward order: 12, 30, 10

Values in reversed order: 10, 30, 12

**4.Write a program to find greatest from three numbers.**

**Code:**

#include <iostream>

using namespace std;

int main() {

double num1, num2, num3;

cout << "Enter the first number: ";

cin >> num1;

cout << "Enter the second number: ";

cin >> num2;

cout << "Enter the third number: ";

cin >> num3;

double greatest;

if (num1 >= num2 && num1 >= num3) {

greatest = num1;

}

else if (num2 >= num1 && num2 >= num3) {

greatest = num2;

}

else {

greatest = num3;

}

cout << "The greatest number is: " << greatest << endl;

return 0;

}

**Output:**

Enter the first number: 22

Enter the second number: 12

Enter the third number: 56

The greatest number is: 56

**5.Write a program in C++ to swap two numbers .**

**Using a Temporary Variable:**

**Code:**

#include <iostream>

using namespace std;

int main() {

int a, b, temp;

cout << "Enter the first number (a): ";

cin >> a;

cout << "Enter the second number (b): ";

cin >> b;

cout << "Before swapping: a = " << a << ", b = " << b << endl;

temp = a;

a = b;

b = temp;

cout << "After swapping: a = " << a << ", b = " << b << endl;

return 0;

}

**Output:**

Enter the first number (a): 20

Enter the second number (b): 30

Before swapping: a = 20, b = 30

After swapping: a = 30, b = 20

**Swapping Without a Temporary Variable**

* using arithmetic operations:

#include <iostream>

using namespace std;

int main() {

int a, b;

cout << "Enter the first number (a): ";

cin >> a;

cout << "Enter the second number (b): ";

cin >> b;

cout << "Before swapping: a = " << a << ", b = " << b << endl;

a = a + b;

b = a - b;

a = a - b;

cout << "After swapping: a = " << a << ", b = " << b << endl;

return 0;

}

**Output:**

Enter the first number (a): 20

Enter the second number (b): 10

Before swapping: a = 20, b = 10

After swapping: a = 10, b = 20

**6.To calculate the area of circle, rectangle and triangle using function overloading.**

**Code:**

#include <iostream>

#include<conio.h>

using namespace std;

int area(int, int);

float area(float);

float area(float, float);

int main() {

int l, b;

float r, bs, ht;

cout << "Enter the length and breadth of rectangle:";

cin >> l >> b;

cout << "Enter the radius of circle:";

cin >> r;

cout << "Enter the base and height of triangle:";

cin >> bs >> ht;

cout << "\nArea of rectangle is " << area(l, b);

cout << "\nArea of circle is " << area(r);

cout << "\nArea of triangle is " << area(bs, ht);

return 0;

}

int area(int l, int b) {

return(l \* b);

}

float area(float r) {

return(3.14 \* r \* r);

}

float area(float bs, float ht) {

return((bs \* ht) / 2);

}

**Output:**

Enter the length and breadth of rectangle:80

20

Enter the radius of circle:10

Enter the base and height of triangle:20

100

Area of rectangle is 1600

Area of circle is 314

Area of triangle is 1000

**7.Write a C++ Program to sort an array of numbers in descending order**

**Code:**

#include <iostream>

using namespace std;

int main()

{

int arr[10] = { 34,86,7,90,63,72,14,81,74,18 };

int n = 10, i, j;

int temp;

cout << "Unsorted Array elements:" << endl;

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

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

}

cout << endl;

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

{

for (j = i + 1; j < n; j++)

{

if (arr[i] < arr[j])

{

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

cout << "Sorted Array elements in descending order:" << endl;

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

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

}

cout << endl;

return 0;

}

**Output:**

Unsorted Array elements:

34 86 7 90 63 72 14 81 74 18

Sorted Array elements in descending order:

90 86 81 74 72 63 34 18 14 7

**8.Write a C++ program to accept and display employee (e\_no,e\_name,e\_designation) details using this pointer**

**Code:**

#include <iostream>

#include <string.h>

using namespace std;

class Employee {

public:

int id;

char name[50];

char designation[50];

Employee(int id, char name[50], char designation[50]) {

this->id = id;

strcpy(this->name, name);

strcpy(this->designation, designation);

}

void display() {

cout << this->id << " " << this->name << " " << this->designation << endl;

}

};

int main() {

int id;

char name[50];

char designation[50];

cout << "\n Enter employee id : ";

cin >> id;

cin.ignore();

cout << "\n Enter employee name : ";

cin.getline(name, 50);

cout << "\n Enter employee designation : ";

cin.getline(designation, 50);

Employee e1(id, name, designation);

e1.display();

return 0;

}

**Output:**

Enter employee id : 3

Enter employee name : Jagruti Shirsath

Enter employee designation : Data Engineer

3 Jagruti Shirsath Data Engineer

**9.Write a C++ program to print area of circle, square and rectangle using inline function.**

**Code:**

#include <iostream>

#include <string.h>

using namespace std;

inline float areaOfCircle(float r) {

return 3.14 \* r \* r;

}

inline float areaOfRectangle(float l, float b) {

return l \* b;

}

inline float areaOfSquare(float s) {

return s \* s;

}

int main() {

cout << "\nArea of circle is : " << areaOfCircle(20);

cout << "\nArea of rectangle is : " << areaOfRectangle(100, 50);

cout << "\nArea of square is : " << areaOfSquare(40);

}

**Output:**

Area of circle is : 1256

Area of rectangle is : 5000

Area of square is : 1600

**10.Write a C++ program to read an integer n and prints the factorial of n.(using recursive function)**

**Code:**

#include <iostream>

#include<conio.h>

#include <string.h>

using namespace std;

int factorial(int n);

int main() {

int n;

cout << "Enter a positive integer: ";

cin >> n;

cout << "Factorial of " << n << " = " << factorial(n);

}

int factorial(int n) {

if (n > 1) {

return n \* factorial(n - 1);

}

else {

return 1;

}

}

**Output:**

Enter a positive integer: 4

Factorial of 4 = 24

**11.Write a C++ program to compute the sum of the specified number of Prime numbers.**

**Code:**

#include<iostream>

using namespace std;

int main() {

int num, i, count, sum = 0, limit;

cout << "\nEnter the maximum limit : ";

cin >> limit;

for (num = 1; num <= limit; num++) {

count = 0;

for (i = 2; i <= num / 2; i++) {

if (num % i == 0) {

count++;

break;

}

}

if (count == 0 && num != 1) {

sum = sum + num;

}

}

cout << "Sum of prime numbers is: " << sum;

}

**Output:**

Enter the maximum limit : 30

Sum of prime numbers is: 129

**12.Write a program in C++ to demonstrate the manipulators: endl, setw and setfill .**

**Code:**

#include <iostream>

#include <iomanip>

using namespace std;

int main()

{

cout << "Displaying integers with formatting:" << endl;

int num1 = 130;

int num2 = 3500;

int num3 = 90;

cout << num1 << " " << num2 << " " << num3 << endl;

cout << "\nUsing setw and setfill:" << endl;

cout << setfill('-') << setw(10) << num1 << setw(10) << num2 << setw(10) << num3 << endl;

cout << setfill(' ') << setw(10) << num1 << setw(10) << num2 << setw(10) << num3 << endl;

cout << "\nUsing endl to end lines:" << endl;

cout << "Line 1" << endl;

cout << "Line 2" << endl;

cout << "Line 3" << endl;

cout << "\nCombining manipulators:" << endl;

cout << setw(15) << setfill('\*') << "Formatted Text" << endl;

cout << setw(10) << setfill('#') << 40 << endl;

return 0;

}

**Output:**

Displaying integers with formatting:

130 3500 90

Using setw and setfill:

-------130------3500--------90

130 3500 90

Using endl to end lines:

Line 1

Line 2

Line 3

Combining manipulators:

\*Formatted Text

########40

**13. Write a C++ program to demonstrate the use of Friend function in class.**

**Code:**

#include <iostream>

using namespace std;

class Rectangle {

private:

int width;

int height;

public:

Rectangle(int w, int h) : width(w), height(h) {}

friend void displayArea(Rectangle rect);

friend void setDimensions(Rectangle& rect, int w, int h);

};

void displayArea(Rectangle rect) {

int area = rect.width \* rect.height;

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

}

void setDimensions(Rectangle& rect, int w, int h) {

rect.width = w;

rect.height = h;

}

int main() {

Rectangle rect(20, 10);

displayArea(rect);

setDimensions(rect, 20, 8);

displayArea(rect);

return 0;

}

**Output:**

Area of the rectangle: 200

Area of the rectangle: 160

**14.Write a C++ program to sort an Array in Ascending order.**

**C++ Program Using Selection Sort**

**Code:**

#include <iostream>

using namespace std;

void selectionSort(int arr[], int n) {

for (int i = 0; i < n - 1; i++) {

int minIndex = i;

for (int j = i + 1; j < n; j++) {

if (arr[j] < arr[minIndex]) {

minIndex = j;

}

}

if (minIndex != i) {

int temp = arr[i];

arr[i] = arr[minIndex];

arr[minIndex] = temp;

}

}

}

void printArray(int arr[], int size) {

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

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

}

cout << endl;

}

int main() {

int arr[] = { 55, 27, 15, 33, 10

};

int n = sizeof(arr) / sizeof(arr[0]);

cout << "Original array: ";

printArray(arr, n);

selectionSort(arr, n);

cout << "Sorted array: ";

printArray(arr, n);

return 0;

}

**Output:**

Original array: 55 27 15 33 10

Sorted array: 10 15 27 33 55

**15.Write a C++ program to create a class Person which contains data members as P\_Name, P\_City, P\_Contact\_Number. Write member functions to accept and display five Persons information.**

**Code:**

#include <iostream>

#include <string>

using namespace std;

class Person {

private:

string P\_Name;

string P\_City;

string P\_Contact\_Number;

public:

void acceptDetails() {

cout << "Enter Name: ";

getline(cin, P\_Name);

cout << "Enter City: ";

getline(cin, P\_City);

cout << "Enter Contact Number: ";

getline(cin, P\_Contact\_Number);

}

void displayDetails() const {

cout << "Name: " << P\_Name << endl;

cout << "City: " << P\_City << endl;

cout << "Contact Number: " << P\_Contact\_Number << endl;

}

};

int main() {

const int numPersons = 10;

Person persons[numPersons];

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

cout << "Enter details for Person " << (i + 1) << ":\n";

persons[i].acceptDetails();

cout << endl;

}

cout << "\nDisplaying details of all persons:\n";

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

cout << "Person " << (i + 1) << ":\n";

persons[i].displayDetails();

cout << endl;

}

return 0;

}

**Output:**

Enter details for Person 1:

Enter Name: Karan

Enter City: Pune

Enter Contact Number: 345960

Enter details for Person 2:

Enter Name: Jay

Enter City: Mumbai

Enter Contact Number: 234509

Enter details for Person 3:

Enter Name: Riya

Enter City: Nashik

Enter Contact Number: 567890

Enter details for Person 4:

Enter Name: Pradip

Enter City: Pune

Enter Contact Number: 703014

Displaying details of all persons:

Person 1:

Name: Karan

City: Pune

Contact Number: 345960

Person 2:

Name: Jay

City: Mumbai

Contact Number: 234509

Person 3:

Name: Riya

City: Nashik

Contact Number: 567890

Person 4:

Name: Pradip

City: Pune

Contact Number: 703014

**16.Write a C++ program to create a class Student which contains data members as Roll\_Number, Stud\_Name, Marks in five subjects. Write member functions to accept Student information. Display all details of student along with a percentage and class obtained depending on percentage. (Use array of objects)**

**Code:**

#include <iostream>

#include <string>

using namespace std;

class Student {

private:

int Roll\_Number;

string Stud\_Name;

float Marks[5];

public:

void acceptDetails() {

cout << "Enter Roll Number: ";

cin >> Roll\_Number;

cin.ignore();

cout << "Enter Student Name: ";

getline(cin, Stud\_Name);

cout << "Enter marks in 5 subjects:\n";

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

cout << "Subject " << (i + 1) << ": ";

cin >> Marks[i];

}

}

float calculateTotalMarks() const {

float total = 0;

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

total += Marks[i];

}

return total;

}

float calculatePercentage() const {

float total = calculateTotalMarks();

return (total / 500) \* 100;

}

string determineClass() const {

float percentage = calculatePercentage();

if (percentage >= 70) {

return "First Class";

}

else if (percentage >= 60) {

return "Second Class";

}

else if (percentage >= 50) {

return "Third Class";

}

else {

return "Fail";

}

}

void displayDetails() const {

cout << "\nRoll Number: " << Roll\_Number << endl; cout << "Student Name: " << Stud\_Name << endl;

cout << "Marks:\n";

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

cout << "Subject " << (i + 1) << ": " << Marks[i] << endl;

}

float percentage = calculatePercentage();

cout << "Percentage: " << percentage << "%" << endl; cout << "Class Obtained: " << determineClass() << endl;

}

};

int main() {

const int numStudents = 3;

Student students[numStudents];

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

cout << "Enter details for Student " << (i + 1) << ":\n";

students[i].acceptDetails();

cout << endl;

}

cout << "\nDisplaying details of all students:\n";

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

cout << "Details of Student " << (i + 1) << ":\n";

students[i].displayDetails();

cout << endl;

}

return 0;

}

**Output:**

Enter details for Student 1:

Enter Roll Number: 1

Enter Student Name: Ram

Enter marks in 5 subjects:

Subject 1: 45

Subject 2: 56

Subject 3: 67

Subject 4: 78

Subject 5: 89

Enter details for Student 2:

Enter Roll Number: 2

Enter Student Name: Jay

Enter marks in 5 subjects:

Subject 1: 43

Subject 2: 54

Subject 3: 56

Subject 4: 76

Subject 5: 90

Enter details for Student 3:

Enter Roll Number: 3

Enter Student Name: Shruti

Enter marks in 5 subjects:

Subject 1: 65

Subject 2: 55

Subject 3: 68

Subject 4: 79

Subject 5: 95

Displaying details of all students:

Details of Student 1:

Roll Number: 1

Student Name: Ram

Marks:

Subject 1: 45

Subject 2: 56

Subject 3: 67

Subject 4: 78

Subject 5: 89

Percentage: 67%

Class Obtained: First Class

Details of Student 2:

Roll Number: 2

Student Name: Jay

Marks:

Subject 1: 43

Subject 2: 54

Subject 3: 56

Subject 4: 76

Subject 5: 90

Percentage: 63.8%

Class Obtained: First Class

Details of Student 3:

Roll Number: 3

Student Name: Shruti

Marks:

Subject 1: 65

Subject 2: 55

Subject 3: 68

Subject 4: 79

Subject 5: 95

Percentage: 72.4%

Class Obtained: First Class

**17.Create a C++ class for a student object with the following attributes—roll no, name, number of subjects, marks of subjects. Write member function for accepting marks and display all information of student along with total and Percentage. Display marklist with Use of manipulators.**

**Code:**

#include <iostream>

#include <iomanip>

#include <string>

using namespace std;

class Student {

private:

int rollNo;

string name;

int numSubjects;

float\* marks;

public:

Student() : rollNo(0), name(""), numSubjects(0), marks(nullptr) {}

~Student()

{

delete[] marks;

}

void acceptDetails() {

cout << "Enter Roll Number: ";

cin >> rollNo;

cin.ignore();

cout << "Enter Name: ";

getline(cin, name);

cout << "Enter Number of Subjects: ";

cin >> numSubjects;

marks = new float[numSubjects];

cout << "Enter marks for " << numSubjects << " subjects:\n";

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

cout << "Subject " << (i + 1) << ": ";

cin >> marks[i];

}

}

float calculateTotal() const {

float total = 0;

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

total += marks[i];

}

return total;

}

float calculatePercentage() const {

return (calculateTotal() / (numSubjects \* 100)) \* 100;

}

void displayDetails() const {

cout << "\nRoll Number: " << rollNo << endl;

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

cout << "Number of Subjects: " << numSubjects << endl;

cout << left << setw(20) << "Subject" << setw(15) << "Marks" << endl; cout << "------------------------------" << endl;

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

cout << left << setw(20) << ("Subject " + to\_string(i + 1)) << setw(15) << marks[i] << endl;

}

cout << "\nTotal Marks: " << calculateTotal() << endl;

cout << "Percentage: " << fixed << setprecision(2) << calculatePercentage() << "%" << endl;

}

};

int main()

{

Student student;

student.acceptDetails();

student.displayDetails();

return 0;

}

**Output:**

Enter Roll Number: 1

Enter Name: Siya

Enter Number of Subjects: 5

Enter marks for 5 subjects:

Subject 1: 78

Subject 2: 87

Subject 3: 89

Subject 4: 90

Subject 5: 56

Roll Number: 1

Name: Siya

Number of Subjects: 5

Subject Marks

------------------------------

Subject 1 78

Subject 2 87

Subject 3 89

Subject 4 90

Subject 5 56

Total Marks: 400

Percentage: 80.00%

**18.Write a class Complex (real, img) along with appropriate constructors. Also write appropriate functions to overload ‘+’ and ‘-’ operator.**

**Code:**

#include <iostream>

using namespace std;

class Complex {

private:

float real;

float img;

public:

Complex() : real(0), img(0) {}

Complex(float r, float i) : real(r), img(i) {}

Complex operator+(const Complex& other) const {

return Complex(real + other.real, img + other.img);

}

Complex operator-(const Complex& other) const {

return Complex(real - other.real, img - other.img);

}

void display() const {

if (img >= 0)

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

else

cout << real << " - " << -img << "i" << endl;

}

};

int main() {

Complex c1(5.5, 2.5);

Complex c2(1.5, 3.5);

cout << "Complex number 1: ";

c1.display();

cout << "Complex number 2: ";

c2.display();

Complex sum = c1 + c2;

Complex diff = c1 - c2;

cout << "Sum: ";

sum.display();

cout << "Difference: ";

diff.display();

return 0;

}

**Output:**

Complex number 1: 5.5 + 2.5i

Complex number 2: 1.5 + 3.5i

Sum: 7 + 6i

Difference: 4 - 1i

**19.Write a C++ program to find volume of cube, cylinder and rectangle using function overloading**

**Code:**

#include <iostream>

#include <cmath>

using namespace std;

float findVolume(float s) {

return s \* s \* s;

}

float findVolume(float r, float h) {

return 3.14f \* r \* r \* h;

}

float findVolume(float l, float w, float h)

{

return l \* w \* h;

}

int main()

{

cout << "Volume of cube is: " << findVolume(10.0f) << endl;

cout << "Volume of cylinder is: " << findVolume(5.0f, 10.0f) << endl; cout << "Volume of rectangular prism is: " << findVolume(20.0f, 10.0f, 10.0f) << endl;

cout << "Press Enter to exit...";

cin.ignore();

cin.get();

return 0;

}

**Output:**

Volume of cube is: 1000

Volume of cylinder is: 785

Volume of rectangular prism is: 2000

Press Enter to exit...

**20. Write a C++ program to find area of triangle, circle, and rectangle using function overloading.**

**Code:**

#include <iostream>

#include <cmath>

using namespace std;

const float PI = 3.14159265358979323846f;

int area(int length, int breadth) {

return length \* breadth;

}

float area(float radius) {

return PI \* radius \* radius;

}

float area(float base, float height) {

return (base \* height) / 2;

}

int main() {

int length, breadth;

float radius, base, height;

cout << "Enter length and breadth of rectangle: ";

cin >> length >> breadth;

cout << "Enter radius of circle: ";

cin >> radius;

cout << "Enter base and height of triangle: ";

cin >> base >> height;

cout << "\nArea of rectangle is: " << area(length, breadth) << endl; cout << "Area of circle is: " << area(radius) << endl;

cout << "Area of triangle is: " << area(base, height) << endl;

return 0;

}

**Output:**

Enter length and breadth of rectangle: 60

20

Enter radius of circle: 10

Enter base and height of triangle: 30

120

Area of rectangle is: 1200

Area of circle is: 314.159

Area of triangle is: 1800

**21. Create a class student containing data members: a. Roll\_no b. name c. marks1, marks2, marks3 Write necessary member functions:**

**1. To accept details of all students**

**2. To display details of one student**

**3. To display details of all students (Use Function overloading).**

**Code:**

#include <iostream>

#include <vector>

#include <string>

using namespace std;

class Student {

private:

int roll\_no;

string name;

float marks1, marks2, marks3;

public:

void acceptDetails() {

cout << "Enter Roll Number: ";

c in >> roll\_no;

cin.ignore();

cout << "Enter Name: ";

getline(cin, name);

cout << "Enter Marks 1: ";

cin >> marks1;

cout << "Enter Marks 2: ";

cin >> marks2;

cout << "Enter Marks 3: ";

cin >> marks3;

}

void displayDetails() const {

cout << "Roll Number: " << roll\_no << endl;

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

cout << "Marks 1: " << marks1 << endl;

cout << "Marks 2: " << marks2 << endl;

cout << "Marks 3: " << marks3 << endl;

}

static void displayDetails(const vector<Student>& students) {

for (const auto& student : students) {

student.displayDetails();

cout << "------------------------" << endl;

}

}

};

int main() {

vector<Student> students;

int num\_students;

cout << "Enter number of students: ";

cin >> num\_students;

cin.ignore();

students.resize(num\_students);

for (int i = 0; i < num\_students; ++i) {

cout << "\nEntering details for student " << (i + 1) << ":" << endl;

students[i].acceptDetails();

}

cout << "\nDisplaying details of all students:" << endl; Student::displayDetails(students);

return 0;

}

**Output:**

Enter number of students: 3

Entering details for student 1:

Enter Roll Number: 1

Enter Name: ABC

Enter Marks 1: 98

Enter Marks 2: 67

Enter Marks 3: 78

Entering details for student 2:

Enter Roll Number: 2

Enter Name: XYZ

Enter Marks 1: 76

Enter Marks 2: 66

Enter Marks 3: 88

Entering details for student 3:

Enter Roll Number: 3

Enter Name: LMN

Enter Marks 1: 57

Enter Marks 2: 75

Enter Marks 3: 92

Displaying details of all students:

Roll Number: 1

Name: ABC

Marks 1: 98

Marks 2: 67

Marks 3: 78

------------------------

Roll Number: 2

Name: XYZ

Marks 1: 76

Marks 2: 66

Marks 3: 88

------------------------

Roll Number: 3

Name: LMN

Marks 1: 57

Marks 2: 75

Marks 3: 92

**22.Create two classes’ dist1 (meters, centimeters) and dist2 (feet, inches). Accept two distances from the user, one in meters and centimeters and the other in feet and inches. Find the sum and difference of the two distances. Display the result in both (meters and centimeters) as well as feet and inches (use friend function).**

**Code:**

#include <iostream>

using namespace std;

class Dist2;

class Dist1 {

private:

float meters;

float centimeters;

public:

Dist1(float m = 0, float cm = 0) : meters(m), centimeters(cm) {}

void acceptDetails() {

cout << "Enter distance in meters: ";

cin >> meters;

cout << "Enter additional centimeters: ";

cin >> centimeters;

normalize();

}

void display() const {

cout << meters << " meters and " << centimeters << " centimeters";

}

Dist2 toDist2() const;

friend void calculateAndDisplay(const Dist1& d1, const Dist2& d2);

private:

void normalize() {

if (centimeters >= 100) {

meters += centimeters / 100;

centimeters = (int)centimeters % 100;

}

}

};

class Dist2 {

private:

float feet;

float inches;

public:

Dist2(float f = 0, float in = 0) : feet(f), inches(in) {}

void acceptDetails() {

cout << "Enter distance in feet: ";

cin >> feet;

cout << "Enter additional inches: ";

cin >> inches;

normalize();

}

void display() const {

cout << feet << " feet and " << inches << " inches";

}

Dist1 toDist1() const;

friend void calculateAndDisplay(const Dist1& d1, const Dist2& d2);

private:

void normalize() {

if (inches >= 12) {

feet += inches / 12;

inches = (int)inches % 12;

}

}

};

Dist2 Dist1::toDist2() const {

float totalInches = meters \* 39.3701f + centimeters \* 0.393701f;

float feet = totalInches / 12;

float inches = totalInches - feet \* 12;

return Dist2(feet, inches);

}

Dist1 Dist2::toDist1() const {

float totalMeters = feet \* 0.3048f + inches \* 0.0254f;

float meters = (int)totalMeters;

float centimeters = (totalMeters - meters) \* 100;

return Dist1(meters, centimeters);

}

void calculateAndDisplay(const Dist1& d1, const Dist2& d2) {

Dist2 d2InFeetInches = d1.toDist2();

Dist1 d1FromD2 = d2.toDist1();

float totalMeters = d1.meters + d1FromD2.meters;

float totalCentimeters = d1.centimeters + d1FromD2.centimeters;

if (totalCentimeters >= 100) {

totalMeters += totalCentimeters / 100;

totalCentimeters = (int)totalCentimeters % 100;

}

float totalFeet = d2.feet + d2InFeetInches.feet;

float totalInches = d2.inches + d2InFeetInches.inches;

if (totalInches >= 12) {

totalFeet += totalInches / 12;

totalInches = (int)totalInches % 12;

}

cout << "Sum in meters and centimeters: " << totalMeters << " meters and " << totalCentimeters << " centimeters" << endl;

cout << "Sum in feet and inches: " << totalFeet << " feet and " << totalInches << " inches" << endl;

float diffMeters = d1.meters - d1FromD2.meters;

float diffCentimeters = d1.centimeters - d1FromD2.centimeters;

if (diffCentimeters < 0) {

diffMeters -= 1;

diffCentimeters += 100;

}

float diffFeet = d2.feet - d2InFeetInches.feet;

float diffInches = d2.inches - d2InFeetInches.inches;

if (diffInches < 0) {

diffFeet -= 1;

diffInches += 12;

}

cout << "Difference in meters and centimeters: " << diffMeters << " meters and " << diffCentimeters << " centimeters" << endl;

cout << "Difference in feet and inches: " << diffFeet << " feet and " << diffInches << " inches" << endl;

}

int main()

{

Dist1 d1;

Dist2 d2;

cout << "Enter details for distance 1 (meters and centimeters):" << endl;

d1.acceptDetails();

cout << "\nEnter details for distance 2 (feet and inches):" << endl;

d2.acceptDetails();

cout << "\nCalculating and displaying results:\n";

calculateAndDisplay(d1, d2);

return 0;

}

**Output:**

Enter details for distance 1 (meters and centimeters):

Enter distance in meters: 1500

Enter additional centimeters: 12

Enter details for distance 2 (feet and inches):

Enter distance in feet: 1000

Enter additional inches: 10

Calculating and displaying results:

Sum in meters and centimeters: 1805 meters and 17.4016 centimeters

Sum in feet and inches: 5921.66 feet and 10.0039 inches

Difference in meters and centimeters: 1195 meters and 6.59839 centimeters

Difference in feet and inches: -3921.66 feet and 9.99609 inches

**23. Write a C++ program using class which contains two data members of type integer. Create and initialize the object using default constructor, parameterized constructor and parameterized constructor with default value. Write a member function to display maximum from given two numbers for all objects**

**Code:**

#include <iostream>

using namespace std;

class NumberPair {

private:

int num1;

int num2;

public:

NumberPair() : num1(0), num2(0) {}

NumberPair(int a, int b = 0) : num1(a), num2(b) {}

void displayMax() const {

int maxNum = (num1 > num2) ? num1 : num2;

cout << "Maximum of " << num1 << " and " << num2 << " is: " << maxNum << endl;

}

};

int main() {

NumberPair np1;

cout << "Object created using default constructor:" << endl;

np1.displayMax();

NumberPair np2(20, 30);

cout << "Object created using parameterized constructor with two arguments:" << endl;

np2.displayMax();

NumberPair np3(15);

cout << "Object created using parameterized constructor with default value:" << endl;

np3.displayMax();

return 0;

}

**Output:**

Object created using default constructor:

Maximum of 0 and 0 is: 0

Object created using parameterized constructor with two arguments:

Maximum of 20 and 30 is: 30

Object created using parameterized constructor with default value:

Maximum of 15 and 0 is: 15

**25.Create a class time that contains hours, minute and seconds as data members. Write the member function to overload operator ‘+’ to add two object of type time,(Use Parameterized constructor to accept values for time). Code:**

#include <iostream>

using namespace std;

class Time {

private:

int hours;

int minutes;

int seconds;

void normalize() {

if (seconds >= 60) {

minutes += seconds / 60;

seconds %= 60;

}

if (minutes >= 60) {

hours += minutes / 60;

minutes %= 60;

}

}

public:

Time(int h = 0, int m = 0, int s = 0) : hours(h), minutes(m), seconds(s) {

normalize();

}

Time operator+(const Time& other) const {

int totalHours = hours + other.hours;

int totalMinutes = minutes + other.minutes;

int totalSeconds = seconds + other.seconds;

Time result(totalHours, totalMinutes, totalSeconds);

return result;

}

void display() const {

cout << hours << " hours, " << minutes << " minutes, " << seconds << " seconds" << endl;

}

};

int main() {

Time time1(1, 60, 20);

Time time2(2, 20, 55);

Time result = time1 + time2;

cout << "Time 1: ";

time1.display();

cout << "Time 2: ";

time2.display();

cout << "Result of addition: ";

result.display();

return 0;

}

**Output:**

Time 1: 2 hours, 0 minutes, 20 seconds

Time 2: 2 hours, 20 minutes, 55 seconds

Result of addition: 4 hours, 21 minutes, 15 seconds

#include <iostream>

#include <iomanip>

#include <memory>

using namespace std;

class Conversion {

public:

virtual void read() = 0;

virtual void convert() = 0;

virtual void display() const = 0;

virtual ~Conversion() {}

};

class Weight : public Conversion {

private:

double gram;

double kilogram;

public:

void read() override {

cout << "Enter weight in grams: ";

cin >> gram;

while (gram < 0) {

cout << "Weight cannot be negative. Enter a valid weight: ";

cin >> gram;

}

}

void convert() override {

kilogram = gram / 1000.0;

}

void display() const override {

cout << fixed << setprecision(2);

cout << "Weight: " << gram << " grams = " << kilogram << " kilograms" << endl;

}

};

class Volume : public Conversion {

private:

double milliliter;

double liter;

public:

void read() override {

cout << "Enter volume in milliliters: ";

cin >> milliliter;

while (milliliter < 0) {

cout << "Volume cannot be negative. Enter a valid volume: ";

cin >> milliliter;

}

}

void convert() override {

liter = milliliter / 1000.0;

}

void display() const override {

cout << fixed << setprecision(2);

cout << "Volume: " << milliliter << " milliliters = " << liter << " liters" << endl;

}

};

class Currency : public Conversion {

private:

double rupees;

double paise;

public:

void read() override {

cout << "Enter amount in rupees: ";

cin >> rupees;

while (rupees < 0) {

cout << "Amount cannot be negative. Enter a valid amount in rupees: ";

cin >> rupees;

}

}

void convert() override {

paise = rupees \* 100.0;

}

void display() const override {

cout << fixed << setprecision(2);

cout << "Currency: " << rupees << " rupees = " << paise << " paise" << endl;

}

};

int main() {

unique\_ptr<Conversion> conv;

int choice;

cout << "Select conversion type:\n";

cout << "1. Weight\n";

cout << "2. Volume\n";

cout << "3. Currency\n";

cout << "Enter your choice (1/2/3): ";

cin >> choice;

switch (choice) {

case 1:

conv = make\_unique<Weight>();

break;

case 2:

conv = make\_unique<Volume>();

break;

case 3:

conv = make\_unique<Currency>();

break;

default:

cout << "Invalid choice!" << endl;

return 1;

}

conv->read();

conv->convert();

conv->display();

// No need to explicitly delete with unique\_ptr

return 0;

}