**DAY-1 CLASS TEST QUESTIONS AND ANSWERS**

**1.Define a Class String. Write overload function = = compare two strings.**

**Input: First string Apple**

**Second string orange**

**Output: Both not equal**

#include <iostream>

#include <cstring>

class String {

private:

char\* str;

public:

String(const char\* s) {

str = new char[strlen(s) + 1];

strcpy(str, s);

}

~String() {

delete[] str;

}

bool operator==(const String& other) const {

return strcmp(str, other.str) == 0;

}

};

int main() {

String str1("Apple");

String str2("orange");

if (str1 == str2) {

std::cout << "Both are equal" << std::endl;

} else {

std::cout << "Both not equal" << std::endl;

}

return 0;

}

**2.Write a program to find area of circle, rectangle and triangle using constructor overloading.**

**Input : a) 3 output : 28.26**

#include <iostream>

#include <cmath>

class AreaCalculator {

private:

double area;

public:

AreaCalculator(double radius) {

area = 3.14 \* radius \* radius;

}

AreaCalculator(double length, double width) {

area = length \* width;

}

AreaCalculator(double base, double height, bool isTriangle) {

if (isTriangle) {

area = 0.5 \* base \* height;

}

}

double getArea() const {

return area;

}

};

int main() {

AreaCalculator circle(3);

std::cout << "Area of Circle: " << circle.getArea() << std::endl;

return 0;

}

**3.Write a C++ program to demonstrate the working of a copy constructor**

**Sample input & output (P1 & P2 are objects)**

**P1.x =10 , p1.y =15**

**P2.x = 10 , p2.y =15**

#include <iostream>

class Point {

public:

int x, y;

// Parameterized Constructor

Point(int x1, int y1) {

x = x1;

y = y1;

}

// Copy Constructor

Point(const Point &p) {

x = p.x;

y = p.y;

}

// Function to display x and y

void display() {

std::cout << "x = " << x << ", y = " << y << std::endl;

}

};

int main() {

// Creating object P1

Point P1(10, 15);

std::cout << "P1: ";

P1.display();

// Creating object P2 using copy constructor

Point P2 = P1;

std::cout << "P2: ";

P2.display();

return 0;

}

**4.Write a Program to print the following pattern**

**Sample Input:**

**Number of rows: 5**

**2**

**4 4**

**16 16 16**

**256 256 2 56 256**

**65536 65536 65536 65536**

#include<iostream>

using namespace std;

int main() {

int n;

cout << "enter no of rows:";

cin >> n;

int value = 2;

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

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

cout << value << " ";

}

cout << "\n";

value = value \* value; // Issue: This squares the value in each iteration.

}

return 0;

}

**5. Write a program in C++ to print a string in reverse using a pointer.**

**Input: Hello**

**Output: olleH**

#include<iostream>

using namespace std;

void reverseString(const char\* str) {

const char\* ptr = str;

while (\*ptr != '\0') {

ptr++;

}

ptr--;

while (ptr >= str) {

cout << \*ptr;

ptr--;

}

cout << endl;

}

int main() {

const char\* str = "Hello";

cout << "Input: " << str << endl;

cout << "Output: ";

reverseString(str);

return 0;

}

**6.Write a C++ Program to check whether a given number is palindrome or not**

**Input : 123321**

**Output : It is a Palindrome**

#include <iostream>

using namespace std;

int main() {

int num, originalNum, reversedNum = 0, remainder;

cout << "Enter a number: ";

cin >> num;

originalNum = num;

while (num != 0) {

remainder = num % 10;

reversedNum = reversedNum \* 10 + remainder;

num /= 10;

}

if (originalNum == reversedNum) {

cout << originalNum << " is a palindrome." << endl;

} else {

cout << originalNum << " is not a palindrome." << endl;

}

return 0;

}

**7.Given an integer array A[ ] consisting of N non-negative integers representing an elevation map, where**

**Examples:**

**Input: a[ ] = {0,1, 0,2,1,0, 1,3,2,1,2,1}**

**Output : 6**

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int trapWater(const vector<int>& heights) {

int n = heights.size();

if (n == 0) return 0;

vector<int> leftMax(n), rightMax(n);

int waterTrapped = 0;

leftMax[0] = heights[0];

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

leftMax[i] = max(leftMax[i - 1], heights[i]);

}

rightMax[n - 1] = heights[n - 1];

for (int i = n - 2; i >= 0; i--) {

rightMax[i] = max(rightMax[i + 1], heights[i]);

}

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

waterTrapped += min(leftMax[i], rightMax[i]) - heights[i];

}

return waterTrapped;

}

int main() {

vector<int> a = {0, 1, 0, 2, 1, 0, 1, 3, 2, 1, 2, 1};

int result = trapWater(a);

cout << "The amount of water trapped is: " << result << endl;

return 0;

}

**8. Write a C++ program to find whether the person is eligible for vote or not. And if that particular person is not eligible, then print how many years are left to be eligible.**

**Input:**

**Enter your age: 7**

**Output:**

**You are allowed to vote after 11 years**

#include <iostream>

using namespace std;

int main() {

int age;

const int votingAge = 18;

cout << "Enter your age: ";

cin >> age;

if (age >= votingAge) {

cout << "You are eligible to vote." << endl;

} else {

int yearsLeft = votingAge - age;

cout << "You are not eligible to vote." << endl;

cout << "You need to wait " << yearsLeft << " more year(s) to be eligible." << endl;

}

return 0;

}

**9. Write a CPP program to find the Square root and Cube root of a number.**

**Square Root Input: 1296**

**Cube root Input: 27**

**Output:**

**36**

**3**

#include <iostream>

#include <cmath>

using namespace std;

int main() {

double number;

cout << "Enter a number: ";

cin >> number;

if (number < 0) {

cout << "Error: Negative number entered. Cannot compute square root or cube root of a negative number." << endl;

} else {

double squareRoot = sqrt(number);

cout << "Square root of " << number << " is " << squareRoot << endl;

double cubeRoot = cbrt(number);

cout << "Cube root of " << number << " is " << cubeRoot << endl;

}

return 0;

}

**10. Write a C++ program that reads an integer and print the least significant digit and the next least significant digit.**

**Example:**

**Input:**

**Enter an integer number : 7235**

**Output:**

**The least significant digit is 5 The next least significant digit is 3**

#include <iostream>

using namespace std;

int main() {

int number;

cout << "Enter an integer: ";

cin >> number;

if (number < 0) {

cout << "Error: Negative number entered. Please enter a non-negative integer." << endl;

return 1;

}

int lsd = number % 10;

int nextLsd = (number / 10) % 10;

cout << "Least significant digit: " << lsd << endl;

cout << "Next least significant digit: " << nextLsd << endl;

return 0;

}

**12.Write a program to print the area and perimeter of a triangle having sides of 3,4 and 5 units by creating a class named ‘ Triangle’ With a function to print the area and perimeter.**

#include <iostream>

#include <cmath>

using namespace std;

class Triangle {

private:

double a, b, c;

public:

Triangle(double side1, double side2, double side3) : a(side1), b(side2), c(side3) {}

void printPerimeter() {

double perimeter = a + b + c;

cout << "Perimeter of the triangle: " << perimeter << endl;

}

void printArea() {

double s = (a + b + c) / 2;

double area = sqrt(s \* (s - a) \* (s - b) \* (s - c));

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

}

};

int main() {

Triangle tri(3, 4, 5);

tri.printPerimeter();

tri.printArea();

return 0;

}

**13.write a program to find the sum of the series 1!/1+2!/2+3!/3+4!/4+5!/5**

**Sample input : 5**

**Sample Output : 34**

#include <iostream>

using namespace std;

long long factorial(int num) {

long long fact = 1;

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

fact \*= i;

}

return fact;

}

int main() {

int n;

cout << "Enter the number of terms: ";

cin >> n;

double sum = 0.0;

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

sum += static\_cast<double>(factorial(i)) / i;

}

cout << "The sum of the series is: " << sum << endl;

return 0;

}

**14. Create a class ‘ Degree’ having a function ‘get degree’ that prints “ I got a Degree”**

**It has tow subclasses namely ‘Undergraduate’ and postgraduate’ each having a function with the same that prints’ I am an Undergraduate’ and “ I am a postgraduate” respectively. Call the function by creating an object of each of the three classes**.

#include <iostream>

using namespace std;

class Degree {

public:

void getDegree() {

cout << "I got a Degree" << endl;

}

};

class Undergraduate : public Degree {

public:

void getDegree() {

cout << "I am an Undergraduate" << endl;

}

};

class Postgraduate : public Degree {

public:

void getDegree() {

cout << "I am a Postgraduate" << endl;

}

};

int main() {

Degree degree;

Undergraduate undergrad;

Postgraduate postgrad;

degree.getDegree();

undergrad.getDegree();

postgrad.getDegree();

return 0;

}

**15.write a C++ Program to display address of each element of an array.**

**Output :**

**Displaing address using arrays;**

#include <iostream>

using namespace std;

int main() {

int arr[] = {10, 20, 30, 40, 50};

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

cout << "Displaying address using arrays:" << endl;

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

cout << "Address of arr[" << i << "] = " << &arr[i] << endl;

}

return 0;

}