**Module 3 Introduction to OOPS Programming.**

**1. Introduction to C++.**

1. **First C++ Program: Hello World**

**#include<iostream>**

**using namespace std;**

**main() {**

**cout<<"Hello, World!";**

**}**

**2. Basic Input/Output**

**#include<iostream>**

**using namespace std;**

**main() {**

**string name;**

**int age;**

**cout << "Enter your name: ";**

**cin >> name;**

**cout << "Enter your age: ";**

**cin >> age;**

**cout<<"Hello "<<name<<",you are "<<age<<" years old."<< endl;**

**}**

1. **POP vs. OOP Comparison Program**

**#include <iostream>**

**using namespace std;**

**//=======================**

**// Procedural Oriented Programming(POP)**

**//=======================**

**void calcArea\_POP() {**

**float length, width, area;**

**cout << "\n---Procedural Oriented Programming(POP)---";**

**cout << "\nEnter length: ";**

**cin >> length;**

**cout << "Enter width: ";**

**cin >> width;**

**area = length \* width;**

**cout << "Area (POP) = " << area << endl;**

**}**

**//=======================**

**//Object-Oriented Programming(OOP)**

**//=======================**

**class Rectangle {**

**float length, width;**

**public:**

**void input() {**

**cout << "\n---Object-Oriented Programming(OOP ---";**

**cout << "\nEnter length: ";**

**cin >> length;**

**cout << "Enter width: ";**

**cin >> width;**

**}**

**void displayArea() {**

**cout << "Area (OOP) = " << (length \* width) << endl;**

**}**

**};**

**main() {**

**// POP Example**

**calcArea\_POP();**

**// OOP Example**

**Rectangle r;**

**r.input();**

**r.displayArea();**

**}**

1. **Setting Up Development Environment.**

**#include <iostream>**

**using namespace std;**

**main(){**

**int num1, num2, sum;**

**cout << "Enter first number: ";**

**cin >> num1;**

**cout << "Enter second number: ";**

**cin >> num2;**

**sum = num1 + num2;**

**cout << "Sum = " << sum << endl;**

**}**

1. Variables, Data Types, and Operators.
2. **Variables and Constants.**

**#include <iostream>**

**using namespace std;**

**main(){**

**// Constant declaration**

**const double PI = 3.14159;**

**// Variable declarations**

**int age = 20;**

**float height =6.0;**

**char grade = 'A';**

**string name = "Patel Om";**

**double radius = 7.5;**

**// Performing operations**

**double area = PI \* radius \* radius;**

**age = age + 2;**

**// Display values**

**cout << "\nName: " << name << endl;**

**cout << "Age: " << age << endl;**

**cout << "Height: " << height << endl;**

**cout << "Grade: " << grade << endl;**

**cout << "Radius: " << radius << endl;**

**cout << "Area of Circle: " << area << endl;**

**cout << "PI (Constant) = " << PI << endl;**

**}**

1. **Type Conversion.**

**#include <iostream>**

**using namespace std;**

**main() {**

**// Implicit type conversion (Type Promotion)**

**int numInt = 10;**

**double numDouble = numInt; // int ? double automatically**

**cout << "--- Implicit Type Conversion ---" << endl;**

**cout << "Integer value: " << numInt << endl;**

**cout << "Converted to double: " << numDouble << endl;**

**// Explicit type conversion (Type Casting)**

**double pi = 3.14159;**

**int intPi = (int)pi; // Old C-style casting**

**int intPi2 = static\_cast<int>(pi); // Modern C++ casting**

**cout << "\n--- Explicit Type Conversion ---" << endl;**

**cout << "Double value: " << pi << endl;**

**cout << "Converted to int (C-style): " << intPi << endl;**

**cout << "Converted to int (static\_cast): " << intPi2 << endl;**

**// Mixed operation showing implicit conversion**

**int x = 5;**

**double y = 2.5;**

**double result = x + y;**

**cout << "\n--- Mixed Operation ---" << endl;**

**cout << "x + y = " << result << endl;**

**}**

1. **Operator Demonstration.**

**#include <iostream>**

**using namespace std;**

**main() {**

**int a = 18, b = 9;**

**// Arithmetic Operators**

**cout << "--- Arithmetic Operators ---" << endl;**

**cout << "a + b = " << (a + b) << endl;**

**cout << "a - b = " << (a - b) << endl;**

**cout << "a \* b = " << (a \* b) << endl;**

**cout << "a / b = " << (a / b) << endl;**

**cout << "a % b = " << (a % b) << endl;**

**// Relational Operators**

**cout << "\n--- Relational Operators ---" << endl;**

**cout << "a == b : " << (a == b) << endl;**

**cout << "a != b : " << (a != b) << endl;**

**cout << "a > b : " << (a > b) << endl;**

**cout << "a < b : " << (a < b) << endl;**

**cout << "a >= b : " << (a >= b) << endl;**

**cout << "a <= b : " << (a <= b) << endl;**

**// Logical Operators**

**bool x = true, y = false;**

**cout << "\n--- Logical Operators ---" << endl;**

**cout << "x && y : " << (x && y) << endl;**

**cout << "x || y : " << (x || y) << endl;**

**cout << "!x : " << (!x) << endl;**

**// Bitwise Operators**

**cout << "\n--- Bitwise Operators ---" << endl;**

**cout << "a & b : " << (a & b) << endl; // AND**

**cout << "a | b : " << (a | b) << endl; // OR**

**cout << "a ^ b : " << (a ^ b) << endl; // XOR**

**cout << "a << 1 : " << (a << 1) << endl; // Left Shift**

**cout << "a >> 1 : " << (a >> 1) << endl; // Right Shift**

**}**

3.Control Flow Statements.

1. **Grade Calculator.**

**#include <iostream>**

**using namespace std;**

**main(){**

**int marks;**

**cout << "Enter marks (0-100): ";**

**cin >> marks;**

**if (marks >= 90 && marks <= 100) {**

**cout << "Grade: A+" << endl;**

**}**

**else if (marks >= 80) {**

**cout << "Grade: A" << endl;**

**}**

**else if (marks >= 70) {**

**cout << "Grade: B" << endl;**

**}**

**else if (marks >= 60) {**

**cout << "Grade: C" << endl;**

**}**

**else if (marks >= 50) {**

**cout << "Grade: D" << endl;**

**}**

**else if (marks >= 0) {**

**cout << "Grade: Fail" << endl;**

**}**

**else {**

**cout << "Invalid marks entered!" << endl;**

**}**

**}**

1. **Number Guessing Game**

**#include <iostream>**

**#include <cstdlib> // For rand() and srand()**

**#include <ctime> // For time()**

**using namespace std;**

**main(){**

**srand(time(0));**

**int secretNumber = rand() % 100 + 1;**

**int guess;**

**int attempts = 0;**

**cout << "=== Number Guessing Game ===" << endl;**

**cout << "Guess the number (1 to 100): " << endl;**

**while (true) {**

**cout << "Enter your guess: ";**

**cin >> guess;**

**attempts++;**

**if (guess == secretNumber) {**

**cout << "Congratulations! You guessed it in " << attempts << " attempts." << endl;**

**break;**

**}**

**else if (guess > secretNumber) {**

**cout << "Too high! Try again." << endl;**

**}**

**else {**

**cout << "Too low! Try again." << endl;**

**}**

**}**

**}**

1. **Multiplication Table**

**#include <iostream>**

**using namespace std;**

**main(){**

**int number;**

**cout<<"Enter a number:";**

**cin>>number;**

**cout<<"\n Multiplication Table of"<<number<<":\n";**

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

**cout<<number<<"x"<<i<<"="<<(number\*i)<<endl;**

**}**

**}**

1. **Nested Control Structures**

**#include <iostream>**

**using namespace std;**

**main(){**

**int rows;**

**cout << "Enter number of rows for the triangle: ";**

**cin >> rows;**

**for (int i = 1; i <= rows; i++){**

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

**cout << "\* ";**

**}**

**cout << endl; // Move to next line after each row**

**}**

**}**

.

4.Functions and Scope.

1. **Simple Calculator Using Functions**

**#include <iostream>**

**using namespace std;**

**// Function declarations**

**float add(float a, float b) { return a + b; }**

**float subtract(float a, float b) { return a - b; }**

**float multiply(float a, float b) { return a \* b; }**

**float divide(float a, float b) { return b != 0 ? a / b : 0; }**

**main() {**

**float num1, num2;**

**char op;**

**cout << "Enter first number: ";**

**cin >> num1;**

**cout << "Enter second number: ";**

**cin >> num2;**

**cout << "Enter operator (+, -, \*, /): ";**

**cin >> op;**

**switch (op) {**

**case '+': cout << "Result: " << add(num1, num2) << endl; break;**

**case '-': cout << "Result: " << subtract(num1, num2) << endl; break;**

**case '\*': cout << "Result: " << multiply(num1, num2) << endl; break;**

**case '/':**

**if (num2 != 0) cout << "Result: " << divide(num1, num2) << endl;**

**else cout << "Error: Division by zero!" << endl;**

**break;**

**default: cout << "Invalid operator!" << endl;**

**}**

**}**

**2. Factorial Calculation Using Recursion.**

**#include <iostream>**

**using namespace std;**

**// Recursive function**

**long long factorial(int n) {**

**if (n <= 1) return 1; // Base case**

**return n \* factorial(n - 1); // Recursive call**

**}**

**main() {**

**int num;**

**cout << "Enter a number: ";**

**cin >> num;**

**if (num < 0)**

**cout << "Factorial Is Not Defined For Negative Numbers." << endl;**

**else**

**cout << "Factorial of " << num << " = " << factorial(num) << endl;**

**}**

1. **Variable Scope**

**#include <iostream>**

**using namespace std;**

**int GlobalVar = 100; // Global variable**

**void displayScope() {**

**int localVar = 50; // Local variable to this function**

**cout<<"Inside function-Global:"<<GlobalVar<<",Local:"<<localVar<< endl;**

**}**

**main(){**

**int localVar = 10; // Local to main()**

**cout<<"Inside main-Global:"<<GlobalVar<<",Local:"<<localVar<< endl;**

**displayScope();**

**}**

1. Arrays and Strings
2. **Array Sum and Average**

**#include <iostream>**

**using namespace std;**

**main(){**

**int n;**

**cout << "Enter number of elements: ";**

**cin >> n;**

**int arr[100]; // Fixed size for simplicity**

**int sum = 0;**

**cout << "Enter " << n << " integers: ";**

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

**cin >> arr[i];**

**sum += arr[i];**

**}**

**double avg = static\_cast<double>(sum) / n;**

**cout << "Sum = " << sum << endl;**

**cout << "Average = " << avg << endl;**

**}**

1. **Matrix Addition**

**#include <iostream>**

**using namespace std;**

**main(){**

**int A[2][2], B[2][2], C[2][2];**

**cout << "Enter elements of first 2x2 matrix:\n";**

**for (int i = 0; i < 2; i++) {**

**for (int j = 0; j < 2; j++) {**

**cin >> A[i][j];**

**}**

**}**

**cout << "Enter elements of second 2x2 matrix:\n";**

**for (int i = 0; i < 2; i++) {**

**for (int j = 0; j < 2; j++) {**

**cin >> B[i][j];**

**}**

**}**

**// Adding matrices**

**for (int i = 0; i < 2; i++) {**

**for (int j = 0; j < 2; j++) {**

**C[i][j] = A[i][j] + B[i][j];**

**}**

**}**

**cout << "Resultant Matrix (A + B):\n";**

**for (int i = 0; i < 2; i++) {**

**for (int j = 0; j < 2; j++) {**

**cout << C[i][j] << " ";**

**}**

**cout << endl;**

**}**

**}**

**3. String Palindrome Check**

**#include <iostream>**

**#include <string>**

**using namespace std;**

**main(){**

**string str;**

**cout << "Enter a string: ";**

**cin >> str;**

**bool isPalindrome = true;**

**int len = str.length();**

**for (int i = 0; i < len / 2; i++) {**

**if (str[i] != str[len - i - 1]) {**

**isPalindrome = false;**

**break;**

**}**

**}**

**if (isPalindrome)**

**cout << "The string is a palindrome." << endl;**

**else**

**cout << "The string is not a palindrome." << endl;**

**}**

1. Introduction to Object-Oriented Programming.
2. **Class for a Simple Calculator**

**#include <iostream>**

**using namespace std;**

**class Calculator {**

**public:**

**double add(double a, double b) {**

**return a + b;**

**}**

**double subtract(double a, double b) {**

**return a - b;**

**}**

**double multiply(double a, double b) {**

**return a \* b;**

**}**

**double divide(double a, double b) {**

**if (b != 0)**

**return a / b;**

**else {**

**cout << "Error! Division by zero." << endl;**

**return 0;**

**}**

**}**

**};**

**main() {**

**Calculator calc;**

**double x, y;**

**cout << "Enter two numbers: ";**

**cin >> x >> y;**

**cout << "Addition: " << calc.add(x, y) << endl;**

**cout << "Subtraction: " << calc.subtract(x, y) << endl;**

**cout << "Multiplication: " << calc.multiply(x, y) << endl;**

**cout << "Division: " << calc.divide(x, y) << endl;**

**}**

1. **Class for Bank Account**

**#include <iostream>**

**using namespace std;**

**class BankAccount {**

**private:**

**double balance;**

**public:**

**BankAccount(double initialBalance) {**

**balance = initialBalance;**

**}**

**void deposit(double amount) {**

**balance += amount;**

**cout << "Deposited: " << amount << endl;**

**}**

**void withdraw(double amount) {**

**if (amount <= balance) {**

**balance -= amount;**

**cout << "Withdrawn: " << amount << endl;**

**} else {**

**cout << "Insufficient balance!" << endl;**

**}**

**}**

**void displayBalance() {**

**cout << "Current Balance: " << balance << endl;**

**}**

**};**

**main() {**

**BankAccount acc(1000);**

**acc.displayBalance();**

**acc.deposit(500);**

**acc.withdraw(300);**

**acc.withdraw(1500);**

**acc.displayBalance();**

**}**

1. **Inheritance Example**

**#include <iostream>**

**using namespace std;**

**class Person {**

**public:**

**string name;**

**int age;**

**void setPersonData(string n, int a) {**

**name = n;**

**age = a;**

**}**

**void displayPersonData() {**

**cout << "Name: " << name << ", Age: " << age << endl;**

**}**

**};**

**class Student : public Person {**

**public:**

**string course;**

**void setStudentData(string n, int a, string c) {**

**setPersonData(n, a);**

**course = c;**

**}**

**void displayStudentData() {**

**displayPersonData();**

**cout << "Course: " << course << endl;**

**}**

**};**

**class Teacher : public Person {**

**public:**

**string subject;**

**void setTeacherData(string n, int a, string s) {**

**setPersonData(n, a);**

**subject = s;**

**}**

**void displayTeacherData() {**

**displayPersonData();**

**cout << "Subject: " << subject << endl;**

**}**

**};**

**main() {**

**Student s;**

**s.setStudentData("Patel Om", 20, "Information Technogoly");**

**s.displayStudentData();**

**Teacher t;**

**t.setTeacherData("Mr. Bob", 45, "Mathematics");**

**t.displayTeacherData();**

**}**