1. First C++ Program: Hello World

o Write a simple C++ program to display "Hello, World!".

#include<iostream>

using namespace std;

int main()

{

cout<<"hello world !";

}

1. Basic Input/Output

o Write a C++ program that accepts user input for their name and age and then displays a personalized greeting.

=>#include<iostream>

using namespace std;

int main()

{

int age;

string name;

cout<<"enter the name : ";

cin>>name;

cout<<"enter the age : ";

cin>>age;

cout<<name<<ends;

cout<<age;

}

#include<stdio.h>

int main()

{

int l,b;

printf("\nenter the length : ");

scanf("%d",&l);

printf("\nenter the width : ");

scanf("%d",&w);

int area = l \* w;

printf("area of rectangle is = %d",area);

}

1. POP vs. OOP Comparison Program o Write two small programs: one using Procedural Programming (POP) to calculate the area of a rectangle, and another using Object-Oriented Programming (OOP) with a class and object for the same task.

#include<iostream>

using namespace std;

class a{

public:

int l,b;

void input()

{

cout<<"enter the length : ";

cin>>l;

cout<<"enter the width : ";

cin>>b;

}

void cal()

{

int area=l\*b;

cout<<area;

}

};

int main()

{

a area;

area.input();

area.cal();

}

1. Variables and Constants o Write a C++ program that demonstrates the use of variables and constants. Create variables of different data types and perform operations on them.

#include<iostream>

using namespace std;

int main()

{

int age=10;

char ch='a';

string s="monank";

float f=95.45;

bool b = true;

const float pi=3.14;

cout<<age<< endl;

cout<<ch<< endl;

cout<<s<< endl;

cout<<f<< endl;

cout<<(b?"yes":"no")<<endl;

int year = 30-age;

float circle = 2\*pi\*5;

cout<<year;

cout<<circle;

}

1. Type Conversion o Write a C++ program that performs both implicit and explicit type conversions and prints the results.

#include<iostream>

using namespace std;

int main()

{

int a=10;

double d = a;

cout<<"impicit type Conversion : \n";

cout<<a<<endl;

cout<<d<<endl;

double pi = 3.14;

int b = pi;

cout<<pi<<endl;

cout<<b<<endl;

cout<<"explicit type Conversion : ";

double num = 99.95;

int n = (int)num;

cout<<num<<endl;

cout<<n;

}

7).Operator Demonstration o Write a C++ program that demonstrates arithmetic, relational, logical, and bitwise operators. Perform operations using each type of operator and display the results.

#include<iostream>

using namespace std;

int main()

{

int a,b,age;

cout<<"enter the number a = ";

cin>>a;

cout<<"enter the number b = ";

cin>>b;

cout<<"enter the age = ";

cin>>age;

int sum = a+b;

cout<<"sum is ="<<sum<<endl;

if(a==b)

{

cout<<"both are same\n";

}

else

{

cout<<"both are different\n";

}

if(age>18&&age<30)

{

cout<<"voting for vote\n";

}

else

{

cout<<"not voting for vote\n";

}

int result = a&b;

cout<<result;

}

8). 1. Grade Calculator o Write a C++ program that takes a student’s marks as input and calculates the grade based on if-else conditions.

#include<iostream>

using namespace std;

int main()

{

int marks;

cout<<"enter the marks : ";

cin>>marks;

if(marks>95&&marks<100)

{

cout<<"A grade..";

}

else if(marks>80&&marks<95)

{

cout<<"B grade..";

}

else if(marks>65&&marks<80)

{

cout<<"C grade..";

}

else

{

cout<<"D grade..";

}

}

9). Number Guessing Game o Write a C++ program that asks the user to guess a number between 1 and 100. The program should provide hints if the guess is too high or too low. Use loops to allow the user multiple attempts.

#include<iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

int main()

{

srand(time(0));

int secretNumber = rand() % 100 + 1;

int num;

do

{

cout<<"guess a number between 1 and 100 : ";

cin>>num;

if(num>secretNumber)

{

cout<<" guess is too high\n";

}

else if(num<secretNumber)

{

cout<<" guess is too low\n";

}

else

{

cout<<"congratulations you are win";

}

}while(num!=secretNumber);

}

10). Multiplication Table o Write a C++ program to display the multiplication table of a given number using a for loop.

#include<iostream>

using namespace std;

int main()

{

int num,i;

cout<<"enter the number of multiplication table : ";

cin>>num;

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

{

cout<<num<<" \* "<<i<<" = "<<num\*i<<endl;

}

}

11). Nested Control Structures o Write a program that prints a right-angled triangle using stars (\*) with a nested loop.

#include <iostream>

using namespace std;

int main() {

int rows;

cout << "Enter the 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;

}

return 0;

}

12).Simple Calculator Using Functions o Write a C++ program that defines functions for basic arithmetic operations (add, subtract, multiply, divide). The main function should call these based on user input.

#include<iostream>

using namespace std;

int sum(int a,int b)

{

int sum=a+b;

return sum;

}

int sub(int a,int b)

{

int sub=a-b;

return sub;

}

int mul(int a,int b)

{

int mul=a\*b;

return mul;

}int div(int a,int b)

{

int div=a/b;

return div;

}

int main()

{

int a,b;

cout<<"enter the number a : ";

cin>>a;

cout<<"enter the number b : ";

cin>>b;

int result = sum(a,b);

int result1 = sub(a,b);

int result2 = mul(a,b);

int result3 = div(a,b);

cout<<"sum is = "<<result<<endl;

cout<<"sub is = "<<result1<<endl;

cout<<"mul is = "<<result2<<endl;

cout<<"div is = "<<result3;

}

12). Factorial Calculation Using Recursion o Write a C++ program that calculates the factorial of a number using recursion.

#include<iostream>

using namespace std;

int fact(int num)

{

if(num!=0)

{

return num \* fact(num-1);

}

else

{

return 1;

}

}

int main()

{

int num;

cout<<"enter the number";

cin>>num;

int r = fact(num);

cout<<r;

}

13). Variable Scope o Write a program that demonstrates the difference between local and global variables in C++. Use functions to show scope.

#include<iostream>

using namespace std;

int globel = 100;

void showglobel()

{

globel+=50;

cout<<globel<<endl;

}

void showlocal()

{

int local = 100;

cout<<local<<endl;

cout<<globel;

}

int main()

{

showglobel();

showlocal();

}

14). Array Sum and Average o Write a C++ program that accepts an array of integers, calculates the sum and average, and displays the results.

#include<iostream>

using namespace std;

int main()

{

int arr[50],size,i,sum=0;

float avg;

cout<<"enter the size : ";

cin>>size;

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

{

cout<<"enter the element : ";

cin>>arr[i];

sum = sum+arr[i];

}

cout<<"sum is = "<<sum<<endl;

cout<<"avg is = "<<(float)sum/(float)size;

}

15). Matrix Addition o Write a C++ program to perform matrix addition on two 2x2 matrices.

#include<iostream>

using namespace std;

int main()

{

int arr[50][50],arr2[50][50],sum[50][50],size,i,k;

float avg;

cout<<"enter the size : ";

cin>>size;

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

{

for(k=0;k<size;k++)

{

cout<<"enter the element a : ";

cin>>arr[i][k];

}

}

cout<<"\n";

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

{

for(k=0;k<size;k++)

{

cout<<"enter the element b : ";

cin>>arr2[i][k];

}

}

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

{

for(k=0;k<size;k++)

{

sum[i][k]=arr[i][k]+arr2[i][k];

}

}

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

{

for(k=0;k<size;k++)

{

cout<<sum[i][k]<<" ";

}

cout<<endl;

}

}

16). String Palindrome Check o Write a C++ program to check if a given string is a palindrome (reads the same forwards and backwards).

#include <iostream>

#include <string>

using namespace std;

bool isPalindrome(string str) {

int start = 0;

int end = str.length() - 1;

while (start < end) {

if (str[start] != str[end]) {

return false;

}

start++;

end--;

}

return true;

}

int main() {

string input;

cout << "Enter a string: ";

cin >> input;

if (isPalindrome(input)) {

cout << "\"" << input << "\" is a palindrome." << endl;

} else {

cout << "\"" << input << "\" is not a palindrome." << endl;

}

return 0;

}

17). Write a C++ program that defines a class Calculator with functions for addition, subtraction, multiplication, and division. Create objects to use these functions.

#include<iostream>

using namespace std;

class calculater{

public:

int a,b;

void input()

{

cout<<"enter the element of a : ";

cin>>a;

cout<<"enter the element of b : ";

cin>>b;

}

void add()

{

int sum = a+b;

cout<<sum<<endl;

}

void sub()

{

int sub = a-b;

cout<<sub<<endl;

}

void mul()

{

int mul = a\*b;

cout<<mul<<endl;

}

void div()

{

int div = a/b;

cout<<div;

}

};

int main()

{

calculater c;

c.input();

c.add();

c.sub();

c.mul();

c.div();

}

#include<iostream>

using namespace std;

18). Class for Bank Account o Create a class BankAccount with data members like balance and member functions like deposit and withdraw. Implement encapsulation by keeping the data members private.

class bankaccount{

private:

int balance;

public:

void deposit(int amount)

{

if(amount>0)

{

balance+=amount;

cout<<"deposit : "<<amount<<endl;

}

else

{

cout<<"invalid deposit amount\n";

}

}

void withdraw(int amount)

{

if(amount>0&&amount<=balance)

{

balance-=amount;

cout<<"withdraw : "<<amount<<endl;

}

else

{

cout<<"invalid withdraw amount\n";

}

}

void currentbalance()

{

cout<<"current balance is : "<<balance<<endl;

}

};

int main()

{

bankaccount b;

b.deposit(500);

b.currentbalance();

b.withdraw(200);

b.currentbalance();

}

19). Inheritance Example o Write a program that implements inheritance using a base class Person and derived classes Student and Teacher. Demonstrate reusability through inheritance.

#include<iostream>

using namespace std;

class person{

public:

void display()

{

cout<<"this is a person class\n";

}

};

class student:public person{

public:

void display2()

{

}

};

class teacher:public person{

public:

void display3()

{

}

};

int main()

{

student s;

s.display();

teacher t;

t.display();

}