**LAB 1:**

**1:write program to test Hello World.**

#include<iostream>

using *namespace* std;

*int* main()

{

    cout<<"Hello World..."<<endl;

    return 0;

}

**Output:**

**Hello World...**

**2:Write a program to adddition of two numbers .**

#include <iostream>

using *namespace* std;

*int* main()

{

*int* num1,num2;

    cout<<"Enter the two number to perform addition:";

    cin>>num1>>num2;

    cout<<"The addition of " << num1 << " and " << num2 << " is " << num1 + num2 << endl;

    return 0;

}

**Output:**

**Enter the two number to perform addition:45 63**

**The addition of 45 and 63 is 108**

**3:Write a program to swap two numbers.**

#include <iostream>

using *namespace* std;

*int* main()

{

*int* num1, num2, temp;

    cout << "Enter the two number to perform Swapping:";

    cin >> num1 >> num2;

    // swaping logic

    cout << "The Number Before Swapping num1=" << num1 << " and num2=" << num2 << endl;

    temp = num1;

    num1 = num2;

    num2 = temp;

    cout << "The Number After Swapping num1=" << num1 << " and num2=" << num2 << endl;

    return 0;

}

**Output:**

**Enter the two number to perform Swapping:12 86**

**The Number Before Swapping num1=12 and num2=86**

**The Number After Swapping num1=86 and num2=12**

**4. Write a program to accept an integer and check if it is even or odd.**

#include<iostream>

using *namespace* std;

*int* main()

{

*int* num;

    cout<<"Enter the number to check it even or odd:";

    cin>>num;

    if(num%2==0)

    {

        cout<<"the number "<<num<<" is even."<<endl;

    }

    else

    {

         cout<<"the number "<<num<<" is odd."<<endl;

    }

    return 0;

}

**Output:**

**Enter the number to check it even or odd:46**

**the number 46 is even.**

**5. Write a program to accept a number and check if it is divisible by 5 and 7.**

#include<iostream>

using *namespace* std;

*int* main()

{

*int* num;

    cout<<"Enter the number to check divisible by 5 and 7:";

    cin>>num;

    if(num%5==0 && num%7==0)

    {

        cout<<"the number "<<num<<" is divisible by 5 and 7.."<<endl;

    }

    else

    {

        cout<<"the number "<<num<<" is not divisible by 5 and 7.."<<endl;

    }

    return 0;

}

**Output:**

**Enter the number to check divisible by 5 and 7:35**

**the number 35 is divisible by 5 and 7..**

**6. Write a program, which accepts annual basic salary of an employee and calculates and displays the**

**Income tax as per the following rules.**

**Basic: < 1, 50,000 Tax = 0**

**1, 50,000 to 3,00,000 Tax = 20%**

**> 3,00,000 Tax = 30%**

#include <iostream>

using *namespace* std;

*int* main()

{

*double* basicsal, tax = 0;

    cout << "Enter the basic salary of employee: ";

    cin >> basicsal;

    if(basicsal<=150000)

    {

        tax=0;

    }

    else if(basicsal>150000 && basicsal<=300000)

    {

        tax=0.2\*(basicsal-150000);

    }

    else

    {

        tax=0.3\*(basicsal-300000)+30000;

    }

    cout << "Income tax of employee is "<<tax<<endl;

    return 0;

}

**Output:**

**Enter the basic salary of employee: 45000**

**Income tax of employee is 0**

**Enter the basic salary of employee: 300000**

**Income tax of employee is 30000**

**7. Accept a lowercase character from the user and check whether the character is a vowel or consonant.**

**(Hint: a, e, i, o, u are vowels)**

#include<iostream>

using *namespace* std;

*int* main()

{

*char* ch='a';

    cout<<"Enter the lower character to check chracter is  vowel or consonant:";

    cin>>ch;

    if(ch=='a'|| ch=='i' || ch=='o' || ch=='e' || ch=='u' )

    {

        cout<<"The character "<<ch<<" is vowel."<<endl;

    }

    else

    {

        cout<<"The character "<<ch<<" is consonant."<<endl;

    }

}

**Output:**

**Enter the lower character to check chracter is vowel or consonant:a**

**The character a is vowel.**

**Enter the lower character to check chracter is vowel or consonant:p**

**The character p is consonant.**

**8. Write a program to input angles of a triangle and check whether triangle is valid or not.**

#include<iostream>

using *namespace* std;

*int* main()

{

*int* a1,a2,a3;

    cout <<"Enter the three angles of the triangle:";

    cin >>a1>>a2>>a3;

    if((a1+a2+a3)== 180)

    {

        cout << "The triangle is valid." << endl;

    }

    else

    {

        cout << "The triangle is not valid." << endl;

    }

    return 0;

}

**Output:**

**Enter the three angles of the triangle:45 90 45**

**The triangle is valid.**

**9:Write a program to find factorial of a given number. ex:no5 fact=5\*4\*3\*2\*1=120**

#include<iostream>

using *namespace* std;

*int* main()

{

*int* num,fact=1;

    cout<<"Enter the number to Fined factorial:";

    cin>>num;

    for(*int* i=1;i<=num;i++)

    {

        fact\*=i;

    }

    cout<<"factorial of number "<<num<<" is "<<fact<<endl;

    return 0;

}

**Output:**

**Enter the number to Fined factorial:6**

**factorial of number 6 is 720**

**10:Write a program to find m to the power n. m=3 and n=4 so 3\*3\*3\*3**

#include<iostream>

using *namespace* std;

*int* main()

{

*int* n,m,res=1;

    cout<<"Enter the value of and m to fined m to power of n.:";

    cin>>m>>n;

    for (*int* i = 0; i < n; ++i)

    {

        res\*= m;

    }

    cout<<"The power of "<<m<<" to "<<n<<" is "<<res<<endl;

    return 0;

}

**Output:**

**Enter the value of and m to fined m to power of n.:3 4**

**The power of 3 to 4 is 81**

**11:Check if number is a prime number or not.:**

#include<iostream>

using *namespace* std;

*int* main()

{

*int* num,flag=1;

    cout<<"Enter the number to check prime or not:";

    cin>>num;

    for(*int* i=2;i<num;i++)

    {

        if(num%i==0)

        {

            flag=0;

            break;

        }

    }

    if(flag==1)

    {

        cout<<"The number "<<num<<" is Prime number."<<endl;

    }

    else

    {

        cout<<"The number "<<num<<" is not Prime number."<<endl;

    }

    return 0;

}

**Output:**

**Enter the number to check prime or not:43**

**The number 43 is Prime number.**

**12:Sum of series :**

**1+2+3+….+n**

#include<iostream>

using *namespace* std;

*int* main()

{

*int* n,sum=0;

    cout<<"Enter the n to fined series upto that:";

    cin>>n;

    for(*int* i=1;i<=n;i++)

    {

        sum+=i;

    }

    cout<<"the sum of Series 1+2+3+.......+"<<n<<" is "<<sum<<endl;

    return 0;

}

**Output:**

**Enter the n to fined series upto that:10**

**the sum of Series 1+2+3+.......+10 is 55**

**13:Check whether the number is palindrome or not?**

//13:Check whether the number is palindrome or not?

#include<iostream>

using *namespace* std;

*int* main()

{

*int* num,rem,revnum=0,temp;

    cout<<"Enter the number to check Number is Palindrome or Not:";

    cin>>num;

    temp=num;

    while(num>0)

    {

        rem=num%10;

        revnum=revnum\*10+rem;

        num=num/10;

    }

    if(temp==revnum)

    {

        cout<<"the number "<<temp<<" is palindrome number..."<<endl;

    }

    else

    {

        cout<<"the number "<<temp<<" is not palindrome number..."<<endl;

    }

    return 0;

}

**Output:**

**Enter the number to check Number is Palindrome or Not:12321**

**the number 12321 is palindrome number...**

**14:Write a program to find sum of all even and odd numbers between 1 to n.**

//14:Write a  program to find sum of all even and odd numbers between 1 to n.

#include<iostream>

using *namespace* std;

*int* main()

{

*int* num,se=0,so=0;

    cout<<"Enter value of n to fined sum of even and odd number:";

    cin>>num;

    for(*int* i=1;i<=num;i++)

    {

        if(i%2==0)

        {

            se+=i;

        }

        else

        {

            so+=i;

        }

    }

    cout<<"The sum of Even number is "<<se<<" and sum of Odd is "<<so<<endl;

    return 0;

}

**Output:**

**Enter value of n to fined sum of even and odd number:18**

**The sum of Even number is 90 and sum of Odd is 81**

**15: Write a program to enter a number and print its reverse.**

//15: Write a  program to enter a number and print its reverse.

#include<iostream>

using *namespace* std;

*int* main()

{

*int* num,rem,revnum=0,temp;

    cout<<"Enter the number to Reverse it:";

    cin>>num;

    temp=num;

    while(num>0)

    {

        rem=num%10;

        revnum=revnum\*10+rem;

        num=num/10;

    }

    cout<<"the number "<<temp<<" after reverse is "<<revnum<<endl;

    return 0;

}

**Output:**

**Enter the number to Reverse it:5632**

**the number 5632 after reverse is 2365**

**16:Write a program to print all Prime numbers between 1 to n.**

// 16:Write a  program to print all Prime numbers between 1 to n.

#include <iostream>

using *namespace* std;

*int* main()

{

*int* n, flag;

    cout << "Enter value of n up to you want to print prime number:";

    cin >> n;

    cout<<"Prime number between 1 to "<<n<<" are:";

    for (*int* i = 2; i < n; i++)

    {

        flag = 1;

        for (*int* j = 2; j < i; j++)

        {

            if (i % j == 0)

            {

                flag = 0;

                break;

            }

        }

        if (flag == 1)

        {

            cout << i << " ";

        }

    }

    return 0;

}

**Output:**

**Enter value of n up to you want to print prime number:50**

**Prime number between 1 to 50 are:2 3 5 7 11 13 17 19 23 29 31 37 41 43 47**

**17:Write a program to check entered number is Armstrong number or not.**

// 17:Write a program to check entered number is Armstrong number or not.

#include <iostream>

using *namespace* std;

*int* main()

{

*int* num, rem, arm = 0, temp;

    cout << "Enter the number to check Number is Armtrong or Not:";

    cin >> num;

    temp = num;

    while (num > 0)

    {

        rem = num % 10;

        arm = arm + (rem \* rem \* rem);

        num = num / 10;

    }

    if (temp == arm)

    {

        cout << "the number " << temp << " is Armstrong number..." << endl;

    }

    else

    {

        cout << "the number " << temp << " is not Armstrong number..." << endl;

    }

    return 0;

}

**Output:**

**Enter the number to check Number is Armstrong or Not:153**

**the number 153 is Armstrong number...**

**18:Write a program to find greatest of three numbers using nested if-else.**

// 18:Write a program to find greatest of three numbers using nested if-else.

#include <iostream>

using *namespace* std;

*int* main()

{

*int* num1, num2, num3;

    cout << "Enter three number to fined greatest: ";

    cin >> num1 >> num2 >> num3;

    if (num1 >= num2)

    {

        if (num1 >= num3)

        {

            cout <<"the number "<< num1 <<" is the greatest." << endl;

        }

        else

        {

            cout <<"the number "<< num3 <<" is the greatest." << endl;

        }

    }

    else

    {

        if (num2 >= num3)

        {

             cout <<"the number "<< num2 <<" is the greatest." << endl;

        }

        else

        {

            cout <<"the number "<< num3 <<" is the greatest." << endl;

        }

    }

    return 0;

}

**Output:**

**Enter three number to fined greatest: 23 56 48**

**the number 56 is the greatest.**

**19:Create menu driven program for Pizza Shop.And display total amount,**

//19:Create menu driven program for Pizza Shop.And display total amount,

#include<iostream>

using *namespace* std;

*int* main()

{

*int* choice=0;

*int* total=0;

    cout<<"1.Menu driven program:";

    do{

    cout<<"\n\n1.hot&spicy veg - 145RS\n2.Sweet chilli chicken - 160Rs\n3.Italian Meat - 370RS \n4.Chilli fries - 120 \nPlease enter your choice:";

    cin>>choice;

    switch (choice)

    {

    case 1:

        total+=145;

        cout<<"\n....thaks for buying hot&spicy veg..";

        break;

    case 2:

        total+=160;

        cout<<"\n....thaks for buying Sweet chilli chicken..";

        break;

    case 3:

        total+=370;

        cout<<"\n...thaks for buying Italian Meat..";

        break;

    case 4:

        total+=120;

        cout<<"\n...thaks for buying Chilli fries ..";

        break;

    case 5:

        cout<<"\nYour total bill is:"<<total;

        break;

    default:

        cout<<"you enter the wrong choice...";

    }

    }while(choice!=5);

    return 0;

}

**Output:**

**1.Menu driven program:**

**1.hot&spicy veg - 145RS**

**2.Sweet chilli chicken - 160Rs**

**3.Italian Meat - 370RS**

**4.Chilli fries - 120**

**Please enter your choice:1**

**....thaks for buying hot&spicy veg..**

**1.hot&spicy veg - 145RS**

**2.Sweet chilli chicken - 160Rs**

**3.Italian Meat - 370RS**

**4.Chilli fries - 120**

**Please enter your choice:2**

**....thaks for buying Sweet chilli chicken..**

**1.hot&spicy veg - 145RS**

**2.Sweet chilli chicken - 160Rs**

**3.Italian Meat - 370RS**

**4.Chilli fries - 120**

**Please enter your choice:3**

**...thaks for buying Italian Meat..**

**1.hot&spicy veg - 145RS**

**2.Sweet chilli chicken - 160Rs**

**3.Italian Meat - 370RS**

**4.Chilli fries - 120**

**Please enter your choice:4**

**...thaks for buying Chilli fries ..**

**1.hot&spicy veg - 145RS**

**2.Sweet chilli chicken - 160Rs**

**3.Italian Meat - 370RS**

**4.Chilli fries - 120**

**Please enter your choice:5**

**Your total bill is:795**

**20:Accept a single digit from the user and display it in words. For example, if digit entered is 9, display Nine.**

// 20:Accept a single digit from the user and display it in words. For example, if digit entered is 9, display Nine.

#include <iostream>

using *namespace* std;

*int* main()

{

*int* digit;

    cout << "Enter a digit to convert into word: ";

    cin >> digit;

    switch (digit)

    {

    case 0:

        cout << "Zero" << endl;

        break;

    case 1:

        cout << "One" << endl;

        break;

    case 2:

        cout << "Two" << endl;

        break;

    case 3:

        cout << "Three" << endl;

        break;

    case 4:

        cout << "Four" << endl;

        break;

    case 5:

        cout << "Five" << endl;

        break;

    case 6:

        cout << "Six" << endl;

        break;

    case 7:

        cout << "Seven" << endl;

        break;

    case 8:

        cout << "Eight" << endl;

        break;

    case 9:

        cout << "Nine" << endl;

        break;

    default:

        cout << "You enter the wrong choice....." << endl;

        break;

    }

    return 0;

}

**Output:**

**Enter a digit to convert into word: 5**

**Five**

**21. Write a program, which accepts two integers and an operator as a character (+ - \* / ), performs the**

**corresponding operation and displays the result.**

#include <iostream>

using *namespace* std;

*int* main()

{

*int* num1, num2, res;

*char* op;

    cout << "Enter two integers: ";

    cin >> num1 >> num2;

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

    cin >> op;

    switch (op)

    {

    case '+':

        res = num1 + num2;

        break;

    case '-':

        res = num1 - num2;

        break;

    case '\*':

        res = num1 \* num2;

        break;

    case '/':

        res = *double*(num1 / num2);

        break;

    default:

        cout<<"You enter Invalid operator." << endl;

    }

    cout <<"After performing opertion result is "<<res<< endl;

    return 0;

}

**Output:**

**Enter two integers: 25 36**

**Enter an operator (+, -, \*, /):+**

**After performing opertion result is 61**

**LAB 2:**

**1:Write a program that accepts numbers continuously as long as the number is positive and prints the**

**sum of the given numbers.**

// 1:Write a program that accepts numbers continuously as long as the number is positive and prints the sum of the given numbers.

#include <iostream>

using *namespace* std;

*int* main()

{

*int* num;

*int* sum = 0;

    do

    {

        cout << "Enter the number (for exit enter negative):";

        cin >> num;

        if (num < 0)

        {

            break;

        }

        sum += num;

    } while (num > 0);

    cout << "Sum of the positive number are:" << sum << endl;

    return 0;

}

**Output:**

**Enter the number (for exit enter negative):56**

**Enter the number (for exit enter negative):32**

**Enter the number (for exit enter negative):69**

**Enter the number (for exit enter negative):-8**

**Sum of the positive number are:157**

**2. Write a program to accept two integers x and n and compute x raised to n.**

//2. Write a program to accept two integers x and n and compute x raised to n.

#include<iostream>

using *namespace* std;

*int* main()

{

*int* x,n,res=1;

    cout<<"Enter the value of x and n and compute x raised to n:";

    cin>>x>>n;

    for(*int* i=0;i<n;i++)

    {

        res\*=x;

    }

    cout<<"the value "<< x<<" raised to "<< n<<" is "<<res<<endl;

    return 0;

}

**Output:**

**Enter the value of x and n and compute x raised to n:5 6**

**the value 5 raised to 6 is 15625**

**3. Write a program to accept a character, an integer n and display the next n characters.**

// 3.Write a program to accept a character, an integer n and display the next n characters.

#include <iostream>

using *namespace* std;

*int* main()

{

*char* ch;

*int* n;

    cout << "Enter a character and integer n: ";

    cin >> ch >> n;

    cout << "Next " << n << " characters after " << ch << " are: ";

    for (*int* i = 0; i <= n; i++)

    {

        cout << (*char*)(ch + i) << " ";

    }

    cout << endl;

    return 0;

}

**Output:**

**Enter a character and integer n: P 5**

**Next 5 characters after P are: P Q R S T U**

**4. Write a program to calculate factorial of a number.**

**For e.g. factorial of 5 = 5! = 5 \*4\*3\*2\*1 = 120**

// 4. Write a program to calculate factorial of a number.

// For e.g. factorial of 5 = 5! = 5 \*4\*3\*2\*1 = 120

#include <iostream>

using *namespace* std;

*int* main()

{

*int* num, fact = 1;

    cout << "Enter the number to fined factorial of it:";

    cin >> num;

    for (*int* i = num; i > 0; i--)

    {

        fact \*= i;

    }

    cout << "the factorial of number " << num << " is "<<fact;

    return 0;

}

**Output:**

**Enter the number to fined factorial of it:5**

**the factorial of number 5 is 120**

**5. Write a program to calculate factors of a given number.**

//5. Write a program to calculate factors of a given number.

#include<iostream>

using *namespace* std;

*int* main()

{

*int* num;

    cout<<"Enter the number to fined factors of it:";

    cin>>num;

    cout<<"the factors of number "<<num<<" are:";

    for(*int* i=1;i<=num;i++)

    {

        if(num%i==0)

        {

            cout<<i<<" ";

        }

    }

    return 0;

}

**Output:**

**Enter the number to fined factors of it:56**

**the factors of number 56 are:1 2 4 7 8 14 28 56**

**6. Accept two numbers and calculate GCD of them.**

//6.Accept two numbers and calculate GCD of them.

#include <iostream>

using *namespace* std;

*int* main()

{

*int* num1,num2,min;

    cout<<"Enter the two number to fined gcd of them:";

    cin>>num1>>num2;

    min=num1;

    if(num1>num2)

        min=num2;

    for(*int* i=min;i>=1;i--)

    {

        if(num1%i==0 && num2%i==0)

        {

             cout << "GCD of " << num1 << " and " << num2 << " is: " <<i<< endl;

            break;

        }

    }

    return 0;

}

**Output:**

**Enter the two number to fined gcd of them:45 60**

**GCD of 45 and 60 is: 15**

**7. Write a menu driven program to do following operations :**

**a) Compute area of circle**

**b) Compute area of rectangle**

**c) Compute area of triangle**

**d) Exit**

**Display menu, ask choice to the user, depending on choice accept the parameters and perform the**

**operation. Continue this process until user selects exit option.**

#include <iostream>

using *namespace* std;

*int* main()

{

*char* choice;

*float* radius, length, width, base, height;

    do

    {

        cout << "\nMenu:\n";

        cout << "a) Compute area of circle\n";

        cout << "b) Compute area of rectangle\n";

        cout << "c) Compute area of triangle\n";

        cout << "d) Exit\n";

        cout << "Enter your choice: ";

        cin >> choice;

        switch (choice)

        {

        case 'a':

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

            cin >> radius;

            cout << "Area of the circle: " << 3.14 \* radius \* radius << endl;

            break;

        case 'b':

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

            cin >> length >> width;

            cout << "Area of the rectangle: " << length \* width << endl;

            break;

        case 'c':

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

            cin >> base >> height;

            cout << "Area of the triangle: " << 0.5 \* base \* height << endl;

            break;

        case 'd':

            cout << "thank you..........." << endl;

            break;

        default:

            cout << "We enter wrong choice select a proper choice......." << endl;

        }

    } while (choice != 'd');

    return 0;

}

**Output:**

**Menu:**

**a) Compute area of circle**

**b) Compute area of rectangle**

**c) Compute area of triangle**

**d) Exit**

**Enter your choice: a**

**Enter the radius of the circle: 4**

**Area of the circle: 50.24**

**Menu:**

**a) Compute area of circle**

**b) Compute area of rectangle**

**c) Compute area of triangle**

**d) Exit**

**Enter your choice: b**

**Enter the length and width of the rectangle: 3 5**

**Area of the rectangle: 15**

**Menu:**

**a) Compute area of circle**

**b) Compute area of rectangle**

**c) Compute area of triangle**

**d) Exit**

**Enter your choice: c**

**Enter the base and height of the triangle: 34 5**

**Area of the triangle: 85**

**Menu:**

**a) Compute area of circle**

**b) Compute area of rectangle**

**c) Compute area of triangle**

**d) Exit**

**Enter your choice: d**

**thank you...........**

**8. Write a program to print all prime numbers between 1 to n**

#include <iostream>

using *namespace* std;

*int* main()

{

*int* n, flag;

    cout << "Enter value of n up to you want to print prime number:";

    cin >> n;

    cout<<"Prime number between 1 to "<<n<<" are:";

    for (*int* i = 2; i < n; i++)

    {

        flag = 1;

        for (*int* j = 2; j < i; j++)

        {

            if (i % j == 0)

            {

                flag = 0;

                break;

            }

        }

        if (flag == 1)

        {

            cout << i << " ";

        }

    }

    return 0;

}

**Output:**

**Enter value of n up to you want to print prime number:50**

**Prime number between 1 to 50 are:2 3 5 7 11 13 17 19 23 29 31 37 41 43 47**

**LAB 3:**

**1:Write a program to create an array of integers and perform following operations on that array like**

**finding the sum, average, maximum and minimum number in that array. Accept the numbers of the**

**array from user.**

#include <iostream>

using *namespace* std;

*int* main()

{

*int* size;

    cout << "how many element you want to add in array:";

    cin >> size;

*int* arr[size];

    // store element of array

    for (*int* i = 0; i < size; i++)

    {

        cout << "Enter the " << i << "th element:";

        cin >> arr[i];

    }

    // traverse the array

    cout << "Array element are:";

    for (*int* i = 0; i < size; i++)

    {

        cout << arr[i] << " ";

    }

    // sort the array

    for (*int* i = 0; i < size - 1; i++)

    {

        for (*int* j = 0; j < size - 1 - i; j++)

        {

            if (arr[j] > arr[j + 1])

            {

*int* temp;

                temp = arr[j];

                arr[j] = arr[j + 1];

                arr[j + 1] = temp;

            }

        }

    }

    // traverse the array

    cout << "\nafter sort Array element are:";

*int* amin, amax, asum = 0, avg = 0;

    for (*int* i = 0; i < size; i++)

    {

        cout << arr[i] << " ";

        asum += arr[i];

    }

    avg = asum / size;

    amin = arr[0];

    amax = arr[size - 1];

    cout << "\n\nSum of the elements: " << asum << endl;

    cout << "Average of the elements: " << avg << endl;

    cout << "Maximum number in the array: " << amax << endl;

    cout << "Minimum number in the array: " << amin << endl;

    return 0;

}

**2: Write a program to Accept a number and display its sum of digits.:ex 568 5+6+8**

//2: Write a program to Accept a number and display its sum of digits.:ex 568    5+6+8

#include<iostream>

using *namespace* std;

*int* main()

{

*int* num,rem,sum=0,temp;

    cout<<"Enter the number to Fined sum of number:";

    cin>>num;

    temp=num;

    while(num>0)

    {

        rem=num%10;

        sum+=rem;

        num=num/10;

    }

    cout<<"The Sum of number "<<temp<<" digit are "<<sum<<endl;

    return 0;

}

**Output:**

**Enter the number to Fined sum of number:45663**

**The Sum of number 45663 digit are 24**

**3:. Write a program to find sum of all even and odd numbers between 1 to n.**

// 3:. Write a  program to find sum of all even and odd numbers between 1 to n.

#include <iostream>

using *namespace* std;

*int* main()

{

*int* num, se = 0, so = 0;

    cout << "Enter value of n to fined sum of even and odd number:";

    cin >> num;

    for (*int* i = 1; i <= num; i++)

    {

        if (i % 2 == 0)

        {

            se += i;

        }

        else

        {

            so += i;

        }

    }

    cout << "The sum of Even number is " << se << " and sum of Odd is " << so << endl;

    return 0;

}

**Output:**

**Enter value of n to fined sum of even and odd number:20**

**The sum of Even number is 110 and sum of Odd is 100**

**4:. Write a program to print all Prime numbers between 1 to n.**

//4:. Write a  program to print all Prime numbers between 1 to n.

#include <iostream>

using *namespace* std;

*int* main()

{

*int* n, flag;

    cout << "Enter value of n up to you want to print prime number:";

    cin >> n;

    cout<<"Prime number between 1 to "<<n<<" are:";

    for (*int* i = 2; i < n; i++)

    {

        flag = 1;

        for (*int* j = 2; j < i; j++)

        {

            if (i % j == 0)

            {

                flag = 0;

                break;

            }

        }

        if (flag == 1)

        {

            cout << i << " ";

        }

    }

    return 0;

}

**Output:**

**Enter value of n up to you want to print prime number:50**

**Prime number between 1 to 50 are:2 3 5 7 11 13 17 19 23 29 31 37 41 43 47**

**5:Write a program to accept array from user .Accept number from user and search number is present in array or not.**

#include<iostream>

using *namespace* std;

*int* main()

{

*int* size,search,sflag=0;

    cout<<"how many element you want to add in array:";

    cin>>size;

*int* arr[size];

    //store element of array

    for(*int* i=0;i<size;i++)

    {

        cout<<"Enter the "<<i<<"th element:";

        cin>>arr[i];

    }

    //traverse the array

    cout<<"Array element are:";

    for(*int* i=0;i<size;i++)

    {

        cout<<arr[i]<<" ";

    }

    //search element in array

    cout<<"\nEnter element which you want to search:";

    cin>>search;

    for(*int* i=0;i<size;i++)

    {

        if(arr[i]==search)

        {

            sflag=1;

            cout<<"The element "<<search<<" is present at "<<i<<"th position"<<endl;

            break;

        }

    }

    if(sflag==0)

    {

        cout<<"The element "<<search<<"is not present in array"<<endl;

    }

    return 0;

}

**Output:**

**how many element you want to add in array:5**

**Enter the 0th element:26**

**Enter the 1th element:35**

**Enter the 2th element:86**

**Enter the 3th element:46**

**Enter the 4th element:12**

**Array element are:26 35 86 46 12**

**Enter element which you want to search:86**

**The element 86 is present at 2th position**

**6:Write a program to print following pattern.**

**\***

**\* \***

**\* \* \***

**\* \* \* \***

**\* \* \* \* \***

#include <iostream>

using *namespace* std;

*int* main()

{

*int* rows;

    cout << "How many number of rows we want to print: ";

    cin >> rows;

    for (*int* i = 0; i <rows; i++)

    {

        for (*int* j = 0; j <= i; j++)

        {

            cout << "\* ";

        }

        cout << endl;

    }

    return 0;

}

**Output:**

**How many number of rows we want to print: 5**

**\***

**\* \***

**\* \* \***

**\* \* \* \***

**\* \* \* \* \***

**7:Write a program to create student class with data members rollno, marks1,mark2,mark3.**

**Accept data (acceptInfo()) and display using display member function.**

**Also display total,percentage and grade.**

#include <iostream>

using *namespace* std;

*class* Student

{

*private:*

*int* rollno;

*float* marks1, marks2, marks3;

*float* total, percentage;

*char* grade;

*public:*

*void* acceptInfo()

    {

        cout << "Enter Roll Number: ";

        cin >> rollno;

        cout << "Enter Marks of Subject 1: ";

        cin >> marks1;

        cout << "Enter Marks of Subject 2: ";

        cin >> marks2;

        cout << "Enter Marks of Subject 3: ";

        cin >> marks3;

    }

*void* calculate()

    {

        total = marks1 + marks2 + marks3;

        percentage = (total / 300) \* 100;

        if (percentage >= 90)

            grade = 'A';

        else if (percentage >= 80)

            grade = 'B';

        else if (percentage >= 70)

            grade = 'C';

        else if (percentage >= 60)

            grade = 'D';

        else if (percentage >= 40)

            grade = 'E';

        else

            grade = 'F';

    }

*void* display()

    {

        cout<<"\nStudent detils are:"<<endl;

        cout << "Roll Number: " << rollno << endl;

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

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

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

        cout << "Total Marks: " << total << endl;

        cout << "Percentage: " << percentage << "%" << endl;

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

    }

};

*int* main()

{

    // create object of student

    Student s;

    // call accept method

    s.acceptInfo();

    // calciulate total percentage grade

    s.calculate();

    // display student details

    s.display();

    return 0;

}

**Output:**

**Enter Roll Number: 25**

**Enter Marks of Subject 1: 86**

**Enter Marks of Subject 2: 89**

**Enter Marks of Subject 3: 88**

**Student detils are:**

**Roll Number: 25**

**Marks in Subject 1: 86**

**Marks in Subject 2: 89**

**Marks in Subject 3: 88**

**Total Marks: 263**

**Percentage: 87.6667%**

**Grade: B**

**LAB 4:**

**1:Write a program to create student class with data members rollno, marks1,mark2,mark3.**

**Accept data (acceptInfo()) and display using display member function.**

**Also display total,percentage and grade.**

#include <iostream>

using *namespace* std;

*class* Student

{

*private:*

*int* rollno;

*float* marks1, marks2, marks3;

*float* total, percentage;

*char* grade;

*public:*

*void* acceptInfo()

    {

        cout << "Enter Roll Number: ";

        cin >> rollno;

        cout << "Enter Marks of Subject 1: ";

        cin >> marks1;

        cout << "Enter Marks of Subject 2: ";

        cin >> marks2;

        cout << "Enter Marks of Subject 3: ";

        cin >> marks3;

    }

*void* calculate()

    {

        total = marks1 + marks2 + marks3;

        percentage = (total / 300) \* 100;

        if (percentage >= 90)

            grade = 'A';

        else if (percentage >= 80)

            grade = 'B';

        else if (percentage >= 70)

            grade = 'C';

        else if (percentage >= 60)

            grade = 'D';

        else if (percentage >= 40)

            grade = 'E';

        else

            grade = 'F';

    }

*void* display()

    {

        cout<<"\nStudent detils are:"<<endl;

        cout << "Roll Number: " << rollno << endl;

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

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

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

        cout << "Total Marks: " << total << endl;

        cout << "Percentage: " << percentage << "%" << endl;

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

    }

};

*int* main()

{

    // create object of student

    Student s;

    // call accept method

    s.acceptInfo();

    // calciulate total percentage grade

    s.calculate();

    // display student details

    s.display();

    return 0;

}

**Output:**

**Enter Roll Number: 25**

**Enter Marks of Subject 1: 36**

**Enter Marks of Subject 2: 85**

**Enter Marks of Subject 3: 75**

**Student detils are:**

**Roll Number: 25**

**Marks in Subject 1: 36**

**Marks in Subject 2: 85**

**Marks in Subject 3: 75**

**Total Marks: 196**

**Percentage: 65.3333%**

**Grade: D**

**1. Create a class Person with data members as name, age, city. Write getters and setters for all the data**

**members. Also add the display function. Create Default and Parameterized constructors. Create the**

**object of this class in main method and invoke all the methods in that class.**

#include <iostream>

#include <string>

using *namespace* std;

*class* Person

{

*private:*

        string name;

*int* age;

        string city;

*public:*

        // Default constructor

        Person()

        {

            this->name = "xyz";

            this->age = 0;

            this->city = "abc";

        }

        // Parameterized constructor

        Person(string *name*, *int* *age*, string *city*)

        {

            this->name = *name*;

            this->age = *age*;

            this->city = *city*;

        }

        // Getter

        string getName()

        {

            return name;

        }

*int* getAge()

        {

            return age;

        }

        string getCity()

        {

            return city;

        }

        // setter

*void* setName(string *name*)

        {

*name* = *name*;

        }

*void* setAge(*int* *age*)

        {

*age* = *age*;

        }

*void* setCity(string *city*)

        {

*city* = *city*;

        }

        // Display

*void* display()

        {

            cout << "Person details are:" << endl;

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

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

            cout << "City:" << city << endl;

        }

};

*int* main()

{

    Person p1;

    cout << "We set data using default constructor:" << endl;

    p1.display();

    Person p2("Pankaj", 30, "pune");

    cout << "\nWe set data using Parameterized constructor:" << endl;

    p2.display();

    Person p3;

    cout << "\nWe set data using Setters:" << endl;

    p3.setName("Dukare");

    p3.setAge(25);

    p3.setCity("nashik");

    p3.display();

    cout << "\nWe gate the data using getters:" << endl;

    cout << "Person details are:" << endl;

    cout << "Name:" << p3.getName() << endl;

    cout << "Age:" << p3.getAge()<< endl;

    cout << "City:" << p3.getCity() << endl;

    return 0;

}

**Output:**

**We set data using default constructor:**

**Person details are:**

**Name:xyz**

**Age:0**

**City:abc**

**We set data using Parameterized constructor:**

**Person details are:**

**Name:Pankaj**

**Age:30**

**City:pune**

**We set data using Setters:**

**Person details are:**

**Name:xyz**

**Age:0**

**City:abc**

**We gate the data using getters:**

**Person details are:**

**Name:xyz**

**Age:0**

**City:abc**

**2. Create a class Date with data members as dd, mm, yy. Write getters and setters for all the data members. Also add the display function. Create Default and Parameterized constructors. Create the**

**object of this class in main method and invoke all the methods in that class.**

#include <iostream>

using *namespace* std;

*class* Date

{

*private:*

*int* dd;

*int* mm;

*int* yy;

*public:*

        // Default constructor

        Date()

        {

            dd = 1;

            mm = 1;

            yy = 2000;

        }

        // Parameterized constructor

        Date(*int* *d*, *int* *m*, *int* *y*)

        {

            dd = *d*;

            mm = *m*;

            yy = *y*;

        }

        // Getter

*int* getDay()

        {

            return dd;

        }

*int* getMonth()

        {

            return mm;

        }

*int* getYear()

        {

            return yy;

        }

        //setter

*void* setDay(*int* *day*)

        {

            dd = *day*;

        }

*void* setMonth(*int* *month*)

        {

            mm = *month*;

        }

*void* setYear(*int* *year*)

        {

            yy = *year*;

        }

        // Display

*void* display()

        {

            cout << "Date: " << dd << "/" << mm << "/" << yy << endl;

        }

};

*int* main()

{

    Date d1;

    cout << "we set Date using default constructor:" << endl;

    d1.display();

    Date d2(15,02, 2000);

    cout << "\nwe set Date using Parameterized constructor:" << endl;

    d2.display();

    Date d3;

    cout<<"\nWe set the date using setters"<<endl;

    d3.setDay(25);

    d3.setMonth(12);

    d3.setYear(2002);

    d3.display();

    cout<<"\nwe get the date using getters:"<<endl;

    cout << "Date: " << d3.getDay() << "/" << d3.getMonth() << "/" <<d3.getYear()<< endl;

    return 0;

}

**Output:**

**we set Date using default constructor:**

**Date: 1/1/2000**

**we set Date using Parameterized constructor:**

**Date: 15/2/2000**

**We set the date using setters**

**Date: 25/12/2002**

**we get the date using getters:**

**Date: 25/12/2002**

**3. Create a class Book with data members as bname,id,author,price. Write getters and setters for all the**

**data members. Also add the display function. Create Default and Parameterized constructors. Create**

**the object of this class in main method and invoke all the methods in that class.**

#include <iostream>

#include <string>

using *namespace* std;

*class* Book

{

*private:*

*int* id;

        string bname;

        string author;

*float* price;

*public:*

        // Default constructor

        Book()

        {

            bname = "";

            id = 0;

            author = "";

            price = 0.0;

        }

        // Parameterized constructor

        Book(*int* *id*,string *name*, string *auth*, *float* *price*)

        {

*id* = *id*;

            bname = *name*;

            author = *auth*;

*price* = *price*;

        }

        // Getter

        string getBname()

        {

            return bname;

        }

*int* getId()

        {

            return id;

        }

        string getAuthor()

        {

            return author;

        }

*float* getPrice()

        {

            return price;

        }

        //setter

*void* setBname(string *name*)

        {

            bname = *name*;

        }

*void* setId(*int* *bookId*)

        {

            id = *bookId*;

        }

*void* setAuthor(string *auth*)

        {

            author = *auth*;

        }

*void* setPrice(*float* *bookPrice*)

        {

            price = *bookPrice*;

        }

        // Display

*void* display()

        {

            cout<<"Book Details are:"<<endl;

             cout << "ID:" << id << endl;

            cout << "Name:" << bname << endl;

            cout << "Author:" << author << endl;

            cout << "Price:" << price << endl;

        }

};

*int* main()

{

    Book b1;

    cout << "\nUsing default constructor:" << endl;

    b1.display();

    Book b2(101,"C programming", "pankaj dukare", 100.67);

    cout << "\nUsing Parameterized constructor:" << endl;

    b2.display();

    cout << endl;

    Book b3;

    b3.setId(102);

    cout<<"\nwe set data using setters:"<<endl;

    b3.setBname("Java Programming");

    b3.setAuthor("Dukare pankaj");

    b3.setPrice(121.99);

    b3.display();

    cout<<"\nWe get the data using getters:"<<endl;

    cout<<"Book Details are:"<<endl;

    cout << "ID:" << b3.getId() << endl;

    cout << "Name:" << b3.getBname() << endl;

    cout << "Author:" <<b3.getAuthor() << endl;

    cout << "Price:" << b3.getPrice() << endl;

    return 0;

}

**4. Create a class Point with data members as x,y. Create Default and Parameterized constructors. Write**

**getters and setters for all the data members. Also add the display function. Create the object of this**

**class in main method and invoke all the methods in that class.**

#include <iostream>

using *namespace* std;

*class* Point

{

*private:*

*float* x;

*float* y;

*public:*

        // Default constructor

        Point()

        {

            x = 0.0;

            y = 0.0;

        }

        // Parameterized constructor

        Point(*float* *xValue*, *float* *yValue*)

        {

            x=*xValue*;

            y=*yValue*;

        }

        // Getter

*float* getX()

        {

            return x;

        }

*float* getY()

        {

            return y;

        }

        //setter

*void* setX(*float* *xValue*)

        {

            x = *xValue*;

        }

*void* setY(*float* *yValue*)

        {

            y = *yValue*;

        }

        // Display

*void* display()

        {

             cout << "Point: (" << x << ", " << y << ")" << endl;

        }

};

*int* main()

{

    Point p1;

    cout << "we create point using default constructor:" << endl;

    p1.display();

    Point p2(4.7,2.8);

    cout << "\nwe create point using Parameterized constructor:" << endl;

    p2.display();

    Point p3;

    cout<<"\nwe set the point data using setters:"<<endl;

    p3.setX(4.2);

    p3.setY(9.3);

    p3.display();

    cout<<"\nwe get point detils using getters:"<<endl;

    cout << "Point: (" << p3.getX() << ", " << p3.getY() << ")" << endl;

    return 0;

}

**Output:**

**we create point using default constructor:**

**Point: (0, 0)**

**we create point using Parameterized constructor:**

**Point: (4.7, 2.8)**

**we set the point data using setters:**

**Point: (4.2, 9.3)**

**we get point detils using getters:**

**Point: (4.2, 9.3)**

**5. Create a class ComplexNumber with data members real, imaginary. Create Default and Parameterized constructors. Write getters and setters for all the data members. Also add the display function. Create the object of this class in main method and invoke all the methods in that class.**

/\*

5. Create a class ComplexNumber with data members real, imaginary.

Create Default and Parameterized constructors.Write getters and setters for all the data members.

 Also add the display function. Create the object of this class in main method and invoke all the methods in that class.

\*/

#include <iostream>

using *namespace* std;

*class* ComplexNumber

{

*private:*

*float* real;

*float* img;

*public:*

        // Default constructor

        ComplexNumber()

        {

            this->real = 0.0;

            this->img = 0.0;

        }

        // Parameterized constructor

        ComplexNumber(*float* *r*, *float* *i*)

        {

            this->real = *r*;

            this->img = *i*;

        }

        //setters

*float* getReal()

        {

            return real;

        }

*float* getImg()

        {

            return img;

        }

        //getters

*void* setReal(*float* *r*)

        {

            this->real = *r*;

        }

*void* setImg(*float* *i*)

        {

             this->img = *i*;

        }

        // Display details

*void* display()

        {

            cout << "Complex Number: " << real << " + " << img << "i" << endl;

        }

};

*int* main() {

    ComplexNumber cn1;

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

    cn1.display();

    ComplexNumber cn2(2.7,8.4);

    cout<<"\nusing Parametries constructor:"<<endl;

    cn2.display();

    ComplexNumber cn3;

    cout<<"\nSet the data using setters:"<<endl;

    cn3.setReal(3.8);

    cn3.setImg(4.9);

    cn3.display();

    cout<<"\nget the data using getter:"<<endl;

    cout << "Complex Number: " << cn3.getReal ()<< " + " <<cn3.getImg()<< "i" << endl;

    return 0;

}

**Output:**

**using default constructor:**

**Complex Number: 0 + 0i**

**using Parametries constructor:**

**Complex Number: 2.7 + 8.4i**

**Set the data using setters:**

**Complex Number: 3.8 + 4.9i**

**get the data using getter:**

**Complex Number: 3.8 + 4.9i**

**LAB 6:**

**1 Solve this.**

**Fresh business scenario to apply inheritance , polymorphism to emp based organization scenario.**

**Create Emp based organization structure --- Emp , Mgr , Worker**

**1.1 Emp state--- id(int), name, deptId , basicSalary(double)**

**Accept all of above in constructor arguments.**

**Methods ---**

**1.2. compute net salary ---ret 0**

**(eg : public double computeNetSalary(){return 0;})**

**1.2 Mgr state ---id,name,basic,deptId , perfBonus**

**Add suitable constructor**

**Methods ----**

**1. compute net salary (formula: basic+perfBonus) -- override computeNetSalary**

**1.3 Worker state --id,name,basic,deptId,hoursWorked,hourlyRate**

**Methods :**

**1. compute net salary (formula: = basic+(hoursWorked\*hourlyRate) --override computeNetSalary**

**2. get hrlyRate of the worker -- add a new method to return hourly rate of a worker.(getter)**

**Create suitable array to store organization details.**

**Provide following options**

**1. Hire Manager**

**I/P : all manager details**

**2. Hire Worker**

**I/P : all worker details**

**3. Display information of all employees net salary (by invoking computeNetSal),**

**4. Exit**

#include <iostream>

using *namespace* std;

*class* Emp

{

*protected:*

*int* id;

*int* deptId;

    string name;

*double* basicSal;

*public:*

    Emp(*int* *i*,*int* *deptId*, string *name*, *double* *basicsal*)

    {

        this->id=*i*;

        this->deptId=*deptId*;

        this->name=*name*;

        this->basicSal=*basicsal*;

    }

    virtual *double* computeNetSalary() =0;

    virtual *void* display()

    {

        cout<<"Employee detils are:"<<endl;

        cout << "ID: " << id << ", Name: " << name << ", Department ID: " << deptId << ", Basic Salary:"<< basicSal;

    }

};

*class* Mgr : *public* Emp

{

*private:*

*double* perfBonus;

*public:*

    Mgr(*int* *id*,*int* *deptId*,string *name*,*double* *basicsal*, *double* *perfBonus*):Emp(*id*, *deptId*, *name*, *basicsal*)

    {

        this->perfBonus=*perfBonus*;

    }

*double* computeNetSalary()

    {

        return basicSal + perfBonus;

    }

*void* display()

    {

        Emp::display();

        cout << ", Performance Bonus: $"<< perfBonus << endl;

    }

};

*class* Worker : *public* Emp

{

*private:*

*double* hoursWorked;

*double* hourlyRate;

*public:*

    Worker(*int* *id*,*int* *deptId*,string *name*,*double* *basicsal*,*double* *hoursWorked*, *double* *hourlyRate*):Emp(*id*, *deptId*, *name*, *basicsal*)

    {

        this->hoursWorked=*hoursWorked*;

        this->hourlyRate=*hourlyRate*;

    }

*double* computeNetSalary()

    {

        return basicSal + (hoursWorked \* hourlyRate);

    }

*double* getHourlyRate() const

    {

        return hourlyRate;

    }

*void* display()

    {

        Emp::display();

        cout << ", Hours Worked: " << hoursWorked << ", Hourly Rate: $" << hourlyRate << endl;

    }

};

*int* main()

{

    const *int* MAX\_EMPLOYEES = 100;

    Emp \*employees[MAX\_EMPLOYEES];

*int* numEmployees = 0;

*int* choice;

    do

    {

        cout << "\nOptions:\n";

        cout << "1. Hire Manager\n";

        cout << "2. Hire Worker\n";

        cout << "3. Display information of all employees\n";

        cout << "4. Exit\n";

        cout << "Enter your choice: ";

        cin >> choice;

        switch (choice)

        {

        case 1:

        {

*int* id, deptId;

            string name;

*double* basicSalary, perfBonus;

            cout << "Enter Manager details:\n";

            cout << "ID: ";

            cin >> id;

            cout << "Name: ";

            cin.ignore();

            getline(cin, name);

            cout << "Department ID: ";

            cin >> deptId;

            cout << "Basic Salary: $";

            cin >> basicSalary;

            cout << "Performance Bonus: $";

            cin >> perfBonus;

            employees[numEmployees++] = new Mgr(id, deptId,name,basicSalary, perfBonus);

            cout<<"Manger Hired Successfully..........."<<endl;

            break;

        }

        case 2:

        {

*int* id, deptId;

            string name;

*double* basicSalary, hoursWorked, hourlyRate;

            cout << "Enter Worker details:\n";

            cout << "ID: ";

            cin >> id;

            cout << "Name: ";

            cin.ignore();

            getline(cin, name);

            cout << "Department ID: ";

            cin >> deptId;

            cout << "Basic Salary: $";

            cin >> basicSalary;

            cout << "Hours Worked: ";

            cin >> hoursWorked;

            cout << "Hourly Rate: $";

            cin >> hourlyRate;

            employees[numEmployees++] = new Worker(id, deptId, name, basicSalary, hoursWorked, hourlyRate);

             cout<<"Worker Hired Successfully..........."<<endl;

            break;

        }

        case 3:

        {

            cout << "Information of all employees:\n";

            for (*int* i = 0; i < numEmployees; ++i)

            {

                employees[i]->display();

            }

            break;

        }

        case 4:

        {

            cout << "Thank you.............\n";

            break;

        }

        default:

            cout << "Invalid choice enter write choice.\n";

        }

    } while (choice != 4);

    for (*int* i = 0; i < numEmployees; ++i)

    {

        delete employees[i];

    }

    return 0;

}

**Options:**

**1. Hire Manager**

**2. Hire Worker**

**3. Display information of all employees**

**4. Exit**

**Enter your choice: 1**

**Enter Manager details:**

**ID: 101**

**Name: pnkaj dukare**

**Department ID: 10**

**Basic Salary: $56000**

**Performance Bonus: $230**

**Manger Hired Successfully...........**

**Options:**

**1. Hire Manager**

**2. Hire Worker**

**3. Display information of all employees**

**4. Exit**

**Enter your choice: 2**

**Enter Worker details:**

**ID: 201**

**Name: prashant dukare**

**Department ID: 50**

**Basic Salary: $45000**

**Hours Worked: 3**

**Hourly Rate: $56**

**Worker Hired Successfully...........**

**Options:**

**1. Hire Manager**

**2. Hire Worker**

**3. Display information of all employees**

**4. Exit**

**Enter your choice: 3**

**Information of all employees:**

**Employee detils are:**

**ID: 101, Name: pnkaj dukare, Department ID: 10, Basic Salary:56000, Performance Bonus: $230**

**Employee detils are:**

**ID: 201, Name: prashant dukare, Department ID: 50, Basic Salary:45000, Hours Worked: 3, Hourly Rate: $56**

**Options:**

**1. Hire Manager**

**2. Hire Worker**

**3. Display information of all employees**

**4. Exit**

**Enter your choice: 4**

**Thank you.............**

**2:Create cpp application for bank account handling.**

**2.1. Create a class BankAccount -- acct no(int),customer name(string),balance(double)**

**Add constr. (2 constrs : first to accept all details )**

**2.2 Add Business logic methods**

**Methods**

**public void withdraw(double amt)**

**public void deposit(double amt)**

**2.3: Create object of account class and test withdraw and deposit methods.**

**---------------------------------------------------------------------------------**

#include <iostream>

#include <string>

using *namespace* std;

*class* BankAccount

{

*private:*

*int* acctNo;

    string customerName;

*double* balance;

*public:*

    BankAccount()

    {

        this->acctNo=123;

        this->customerName="xyz";

        this->balance=0.0;

    }

    BankAccount(*int* *acctNo*,string *customerName*, *double* *balance*)

    {

        this->acctNo=*acctNo*;

        this->customerName=*customerName*;

        this->balance=*balance*;

    }

    // Method to withdraw money

*void* withdraw(*double* *amt*)

    {

        if (*amt* > balance)

        {

            cout << "Insufficient funds! Cannot withdraw." << endl;

        }

        else

        {

            balance -= *amt*;

            cout << "Withdrawal successful. Current balance: $" << balance << endl;

        }

    }

    // Method to deposit money

*void* deposit(*double* *amt*)

    {

        balance += *amt*;

        cout << "Deposit successful. Current balance: $" << balance << endl;

    }

    // Getter methods

*int* getAcctNo()

    {

        return acctNo;

    }

    string getCustomerName()

    {

        return customerName;

    }

*double* getBalance()

    {

        return balance;

    }

};

*int* main()

{

    BankAccount account(173, "pankaj dukare", 1000.0);

    cout << "Account Details:" << endl;

    cout << "Account Number: " << account.getAcctNo() << endl;

    cout << "Customer Name: " << account.getCustomerName() << endl;

    cout << "Current Balance: $" << account.getBalance() << endl;

    cout << "\nTesting withdrawal of $500..." << endl;

    account.withdraw(500.0);

    cout << "\nTesting deposit of $2000..." << endl;

    account.deposit(2000.0);

    return 0;

}

**Output:**

**Account Details:**

**Account Number: 173**

**Customer Name: pankaj dukare**

**Current Balance: $1000**

**Testing withdrawal of $500...**

**Withdrawal successful. Current balance: $500**

**Testing deposit of $2000...**

**Deposit successful. Current balance: $2500**

**3:Create a abstract class Shape with pure virtual method area;**

**Create Rectangle,Circle,Square class..inherit them from Shape class..Override area method.**

**Test these all classes by creating object of respective class.**

/\*

3:Create a abstract class Shape with pure virtual method area;

Create Rectangle,Circle,Square class..inherit them from Shape class..Override area method.

Test these all classes by creating object of respective class.

\*/

#include <iostream>

using *namespace* std;

*class* Shape

{

*public:*

    virtual *double* area() = 0;

};

*class* Rectangle : *public* Shape

{

*private:*

*double* length;

*double* width;

*public:*

    Rectangle(*double* *l*, *double* *w*)

    {

        this->length=*l*;

        this->width=*w*;

    }

*double* area ()

    {

        return length \* width;

    }

};

*class* Circle : *public* Shape

{

*private:*

*double* radius;

*public:*

    Circle(*double* *r*)

    {

        this->radius=*r*;

    }

*double* area()

    {

        return 3.14 \* radius \* radius;

    }

};

*class* Square : *public* Shape

{

*private:*

*double* side;

*public:*

    Square(*double* *s*)

    {

        this->side=*s*;

    }

*double* area()

    {

        return side \* side;

    }

};

*int* main()

{

    Rectangle r(5, 3);

    cout << "Area of Rectangle: " << r.area() << endl;

    Circle c(4);

    cout << "Area of Circle: " << c.area() << endl;

    Square s(6);

    cout << "Area of Square: " << s.area() << endl;

    return 0;

}

**Outpute**

**Area of Rectangle: 15**

**Area of Circle: 50.24**

**Area of Square: 36**

**ADDITIONAL PROGRAM**

**1. Opertor Overloading**

#include<iostream>

using *namespace* std;

*class* Box

{

*private:*

*int* l,b,h;

*public:*

    Box()

    {

        cout<<"this is box default constructor"<<endl;

        l=0;

        b=0;

        h=2;

    }

    Box(*int* *l*,*int* *b*,*int* *h*)

    {

        this->l=*l*;

        this->b=*b*;

        this->h=*h*;

    }

*void* acceptdata(*int* *l*,*int* *b*,*int* *h*)

    {

        this->l=*l*;

        this->b=*b*;

       this->h=*h*;

    }

    Box operator+(Box &*obj*)

    {

*int* l,b,c;

        l=this->l+*obj*.l;

        b=this->b+*obj*.b;

        h=this->h+*obj*.h;

        Box b4(l,b,h);

        return b4;

    }

*void* display()

    {

        cout<<"the box details are l="<<l<<" b="<<b<<" h="<<h<<endl;

    }

};

*int* main()

{

    cout<<"opertr overloading"<<endl;

    cout<<"we create a box 1 object:"<<endl;

    Box b1;

    b1.display();

    cout<<"we create a box 2 object:"<<endl;

    Box b2(12,56,23);

    b2.display();

    cout<<"we create a box 3 object:"<<endl;

    Box b3;

    b3.acceptdata(12,45,23);

    b3.display();

    cout<<"\n the details of box b2 and b3"<<endl;

    b2.display();

    b3.display();

    Box b4=b2+b3;

    b4.display();

}

**Output:**

**operter overloading**

**we create a box 1 object:**

**this is box default constructor**

**the box details are l=0 b=0 h=2**

**we create a box 2 object:**

**the box details are l=12 b=56 h=23**

**we create a box 3 object:**

**this is box default constructor**

**the box details are l=12 b=45 h=23**

**the details of box b2 and b3**

**the box details are l=12 b=56 h=23**

**the box details are l=12 b=45 h=23**

**the box details are l=24 b=101 h=46**

**2. file operation:**

#include<iostream>

#include<fstream>

#include<string.h>

using *namespace* std;

*class* Student

{

*private:*

*int* id;

*char* name[20];

*public:*

    Student()

    {

        this->id=1;

        strcpy(this->name,"pankaj");

    }

    Student(*int* *id*,*char* \**nm*)

    {

        this->id=*id*;

       strcpy(this->name,*nm*);

    }

*void* display()

    {

        cout<<"the student details are id="<<id<<" name="<<name<<endl;

    }

};

*int* main()

{

    ofstream o("abca.txt",ios::binary | ios::app);

    cout<<"this is file opertion program:"<<endl;

    cout<<"enter the student detais lik id name:";

*int* id;

*char* name[20];

    cin>>id;

    gets(name);

   Student s1(id,name);

   s1.display();

    o.write((*char*\*)&s1,sizeof(s1));

    o.close();

    cout<<"\n read data from file:";

    Student s2;

    ifstream i("abca.txt",ios::binary);

    while(i.read((*char*\*)&s2,sizeof(s2)))

    {

            s2.display();

    }

    i.close();

    return 0;

}

**Output:**

**this is file opertion program:**

**enter the student detais lik id name:1 pankaj**

**the student details are id=1 name= pankaj**

**read data from file:the student details are id=1 name= pankaj**

**3.Template class**

#include <iostream>

using *namespace* std;

*class* Date

{

*private:*

*int* dd, mm, yyyy;

*public:*

    Date()

    {

        dd = 01,

        mm = 01,

        yyyy = 2000;

    }

    Date(*int* *d*, *int* *m*, *int* *y*)

    {

        dd = *d*;

        mm = *m*;

        yyyy = *y*;

    }

*void* display()

    {

        cout << "Date:" << dd << "/" << mm << "/" << yyyy << endl;

    }

*void* swapObj(Date &*d*)

    {

        Date a;

        a = \*this;

        Date temp;

        temp = a;

        a = *d*;

*d* = temp;

        \*this = a;

        /\*

                cout<<"current:"<<temp1.dd;

                int temp;

                //change the day

                temp=this->dd;

                this->dd=d.dd;

                d.dd=temp;

                //change month

                temp=this->mm;

                this->mm=d.mm;

                d.mm=temp;

                //change year

                temp=this->yyyy;

                this->yyyy=d.yyyy;

                d.yyyy=temp;\*/

    }

};

*class* Book

{

*private:*

*int* id;

    string bname, bauthor;

*public:*

    Book()

    {

        id = 101;

        bname = "java programming";

        bauthor = "Pankaj dukare";

    }

    Book(*int* *id*, string *bname*, string *bauthor*)

    {

        this->id = *id*;

        this->bname = *bname*;

        this->bauthor = *bauthor*;

    }

*void* display()

    {

        cout << "book details are: id=" << id << " name=" << bname << " Author=" << bauthor << endl;

    }

*void* swapObj(Book &*d*)

    {

        Book a, temp;

        a = \*this;

        temp = a;

        a = *d*;

*d* = temp;

        \*this = a;

        /\*

                cout<<"current:"<<temp1.dd;

                int temp;

                //change the day

                temp=this->dd;

                this->dd=d.dd;

                d.dd=temp;

                //change month

                temp=this->mm;

                this->mm=d.mm;

                d.mm=temp;

                //change year

                temp=this->yyyy;

                this->yyyy=d.yyyy;

                d.yyyy=temp;\*/

    }

};

*template* <*class* B>

*void* SwapOData(B *obj1*, B *obj2*)

{

    // date after swaping

    cout << "Object before swaping are:\n Object 1=";

*obj1*.display();

    cout << " object 2 =";

*obj2*.display();

    // call a swap method

*obj1*.swapObj(*obj2*);

    // date after swaping

    cout << "Object after swaping are:\n Date 1=";

*obj1*.display();

    cout << " object 2 =";

*obj2*.display();

}

*template* <*class* T>

*void* swapdata(T *a*, T *b*)

{

    cout << "\nelement before swaping first=" << *a* << " second=" << *b* << endl;

    T temp;

    temp = *a*;

*a* = *b*;

*b* = temp;

    cout << "element before swaping first=" << *a* << " second=" << *b* << endl;

}

*int* main()

{

    // swap premitive type

    cout << "Swap two data with help of templte:";

    swapdata(1, 2);               // pass integer data

    swapdata("pankaj", "dukare"); // pass String data

    swapdata(10.24, 78.97);       // pass a float data

    // swap user define type

    // create two date

    cout << "\n Date example:" << endl;

*int* d, m, y;

    cout << "Enter the first date details day month year:";

    cin >> d >> m >> y;

    Date d1(d, m, y);

    // Date d1(12,12,2012);

    cout << "Enter the Second date details day month year:";

    cin >> d >> m >> y;

    Date d2(d, m, y);

    // Date d2(15,10,2016);

    //  call swap object function

    SwapOData(d1, d2);

    cout << "\n Book example:" << endl;

    // create two book

*int* id;

    string bname, bauthor;

    cout << "Enter the first Book details id name author:";

    cin >> id >> bname >> bauthor;

    // Book b1(101,"C programing","dukare");

    Book b1(id, bname, bauthor);

    cout << "Enter the Second Book details id name author:";

    cin >> id >> bname >> bauthor;

    // Book b1(101,"C programing","dukare");

    Book b2(id, bname, bauthor);

    // Book b2(102,"python programing","pankaj");

    //  call swap object function

    SwapOData(b1, b2);

    return 0;

}

**Output:**

**Swap two data with help of templte:**

**element before swaping first=1 second=2**

**element before swaping first=2 second=1**

**element before swaping first=pankaj second=dukare**

**element before swaping first=dukare second=pankaj**

**element before swaping first=10.24 second=78.97**

**element before swaping first=78.97 second=10.24**

**Date example:**

**Enter the first date details day month year:12 10 2005**

**Enter the Second date details day month year:23 11 2006**

**Object before swaping are:**

**Object 1=Date:12/10/2005**

**object 2 =Date:23/11/2006**

**Object after swaping are:**

**Date 1=Date:23/11/2006**

**object 2 =Date:12/10/2005**

**Book example:**

**Enter the first Book details id name author:1 java pankaj**

**Enter the Second Book details id name author:2 c dukare**

**Object before swaping are:**

**Object 1=book details are: id=1 name=java Author=pankaj**

**object 2 =book details are: id=2 name=c Author=dukare**

**Object after swaping are:**

**Date 1=book details are: id=2 name=c Author=dukare**

**object 2 =book details are: id=1 name=java Author=pankaj**