**SAVEETHA SCHOOL OF ENGINEERING**

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES**

**DEPARTMENT OF DATA SCIENCE**

**INSTITUTE OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Course**  **Code** | **DSA01** | **Course Title** | | **Object Oriented Programming with C++** | |
| **Branch** | **B.Tech-**  **CSE/IT/AI&DS/AI&ML** | | **Year\Academic year** | | **I Year/2022** |

**DAY - 3 - Practice Programs**

1. Read the salary and deductions from user and calculate the Gross Pay, net salary and income tax with various income tax slabs. The salary class will read all details then the deduction class and incometax class will inherit the salary class to perform the above.

**Code:**

#include<iostream> using namespace std;

class salary

{

public:

int bs,net,hra,oa,alla,gp;

salary()

{

cout<<"Enter basic salary : ";

cin>>bs;

cout<<"Enter HRA : ";

cin>>hra;

cout<<"Enter other allowence : ";

cin>>oa;

cout<<"Enter all allowance for net salary : ";

cin>>alla; net=bs+hra+oa+alla;

cout<<"Your net salary is "<<net<<"\n";

gp=bs+hra+oa;

cout<<"Your gross pay is "<<gp<<"\n";

}

};

class deduction

{ public:

void deduction1()

{

cout<<"\n"<<"Income tax calculation Enter your details below

"<<"\n";

}

};

class incometax : public salary

{

public:

incometax()

{

if(net>200000 and net<500000)

{

cout<<"Your income tax is "<<net\*0.5;

}

else if(net>500000 and net<700000)

{

cout<<"Your income tax is "<<(net\*0.10)+12500;

}

else if(net>700000 and net<1000000)

{

cout<<"Your income tax is "<<(net\*0.15)+37500;

}

else if(net>1000000)

{

cout<<"Your income tax is "<<(net\*0.20)+75000;

}

else

{

cout<<"No income tax";

}

}

};

int main()

{

salary p1; deduction p2; p2.deduction1();

incometax p3;

}

**OUTPUT:**

Enter basic salary : 30000

Enter HRA : 4000

Enter other allowence : 3000

Enter all allowance for net salary : 5000

Your net salary is 42000

Your gross pay is 37000

2. Implement a program of maintaining banking account information system using multiple inheritance in C++ Programming. Here class savings derived from class account and class user. Use appropriate functions and variables.

**Code** :

#include<iostream> using namespace std; int bal=0;

class account

{

public: string name,acc,acty;

void getdet()

{

cout<<"\nenter your name : ";

cin>>name;

cout<<"\nenter your account number : ";

cin>>acc;

cout<<"\nenter your account type : ";

cin>>acty;

}

void displaya()

{

cout<<"\nname : "<<name; cout<<"\naccount number : "<<acc;

cout<<"\naccount type : "<<acty;

}

};

class withdraw

{

public:

int amt;

void getwith()

{

cout<<"\nenter the withdrawl amount : ";

cin>>amt;

}

void displayb()

{

if(amt<=bal){

cout<<"\nbalance : "<<bal-amt; cout<<"\nTake your cash !!!";

}

else

cout<<"\ninsufficient balance !!!";

}

};

class savings:public account,public withdraw

{ public:

int amt2; void getdep()

{

cout<<"\nenter the deposit amount : ";

cin>>amt2;

}

void displayc()

{

if(amt2>0){

cout<<"\nbalance : "<<bal+amt2;

cout<<"\ndeposited successfully !!!!";

}

else

cout<<"\ninvalid amount !!!";

}

};

int main()

{

int n; savings s;

s.getdet();

cout<<"\n1.withdraw \n2.deposit \nenter your choice : ";

cin>>n;

// s.displaya(); if(n==1){

s.getwith(); //s.displayb();

}

else if(n==2){

s.getdep(); //s.displaya(); s.displayc();

}

return 0;

}

**OUTPUT:**

enter your name : rahul

enter your account number : 567845362

enter your account type : savings

1.withdraw

2.deposit

enter your choice : 2

enter the deposit amount : 1000

balance : 1000

deposited successfully !!!!

3. Consider an example of declaring the examination result. Design 3 classes student,exam,result. The student class has data members such as that reperesenting number, name of student ,create the class exam,which contains data members reperesenting name of subject,minmum marks,maximum marks, obtained marks for 3 subject derive class result from both student and exam classes. Test the result class in main function ?

**Code** :

#include<iostream> using namespace std; class student

{

public: string name,regno; void getdet()

{

cout<<"\nenter your name : ";

cin>>name;

cout<<"\nenter your reg no : ";

cin>>regno;

}

void displaya()

{

cout<<"\nname : "<<name; cout<<"\nreg no : "<<regno;

}

};

class exam

{

public: string sub1,sub2,sub3; float max=100,min=50,m1,m2,m3; void getmark()

{

cout<<"\nenter the subject 1 : ";

cin>>sub1;

cout<<"\nenter the marks obtained in subject 1: ";

cin>>m1;

cout<<"\nenter the subject 2 : ";

cin>>sub2;

cout<<"\nenter the marks obtained in subject 2 : ";

cin>>m2;

cout<<"\nenter the subject 3 : ";

cin>>sub3;

cout<<"\nenter the marks obtained in subject 3 : ";

cin>>m3;

}

};

class result:public student,public exam

{

public: void displayb()

{

cout<<"\n\n\n"<<sub1<<" : "; cout<<"\nminimum marks : "<<min; cout<<"\nmaximum marks : "<<max; cout<<"\nmarks obtained in "<<sub1<<" : "<<m1; cout<<"\n\n\n"<<sub2<<" : ";

cout<<"\nminimum marks : "<<min; cout<<"\nmaximum marks : "<<max; cout<<"\nmarks obtained in "<<sub2<<" : "<<m2;

cout<<"\n\n\n"<<sub3<<" : "; cout<<"\nminimum marks : "<<min; cout<<"\nmaximum marks : "<<max; cout<<"\nmarks obtained in "<<sub3<<" : "<<m3;

}

};

int main()

{

result r;

r.getdet();

r.getmark();

r.displaya();

r.displayb(); return 0;

}

**OUTPUT:**

enter your name : rohith

enter your reg no : 192111567

enter the subject 1 : maths

enter the marks obtained in subject 1: 98

enter the subject 2 : physics

enter the marks obtained in subject 2 : 87

enter the subject 3 : english

enter the marks obtained in subject 3 : 89

name : rohith

reg no : 192111567

maths :

minimum marks : 50

maximum marks : 100

marks obtained in maths : 98

physics :

minimum marks : 50

maximum marks : 100

marks obtained in physics : 87

english :

minimum marks : 50

maximum marks : 100

marks obtained in english : 89

1. Find the max of an integral data set. The program will ask the user to input the number of data values in the set and each value. The program prints on screen a pointer that points to the max value.

**CODE:**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int n;int i; int max=0;**

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

**cin>>n;**

**cout<<"Enter values in array:\n";**

**int arr[n];**

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

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

**}**

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

**if (arr[u]>max)**

**max=arr[u];**

**}**

**int \*pointer= &max;**

**cout<<"Largest integer value in the array is "<<\*pointer;**

**return 0;**

**}**

**OUTPUT:**

Enter number of values:4

Enter values in array:

23

56

87

56

Largest integer value in the array is 87

1. The user to enter integers as inputs to be stored in the variables 'a' and 'b' respectively. There are also two integer pointers named ptrA and ptrB. Assign the values of 'a' and 'b' to ptrA and ptrB respectively, and display them.

**Code :**

#include<iostream> using namespace std;

void justdisplay(int\*a,int\*b)

{

cout<<"entered A => "<<\*a; cout<<"\nentered B => "<<\*b;

}

int main()

{

int a,b; cout<<"ENTER A => "; cin>>a;

cout<<"ENTER B => "; cin>>b;

justdisplay(&a,&b);

}

**OUTPUT:**

Enter number of values:3

Enter values in array:

45

67

87

Largest integer value in the array is 87

1. Program to read and display the Names, Roll No., Marks and grades of 3 students who have appeared in the examination. Declare the class of name, Roll No. and grade. Create an array of objects to read and display the contents.

**CODE:**

#include<iostream> using namespace std; class name

{

public:

string name; void getname()

{

cout<<"\nenter the name : ";

cin>>name;

}

void displaya()

{

cout<<"\nname : "<<name;

}

}; class roll

{

public: string r;

void getrn()

{

cout<<"\nenter the roll no : ";

cin>>r;

}

void displayb()

{

cout<<"\nroll no : "<<r;

}

};

class grade:public name,public roll

{

public: char g;

void getgra()

{

cout<<"\nenter the grade : "; cin>>g;

}

void displayc()

{

cout<<"\ngrade : "<<g;

}

};

int main()

{

int n,i; grade d[10];

cout<<"\nenter the no of students : "; cin>>n;

for(i=0;i<n;i++){ d[i].getname(); d[i].getrn(); d[i].getgra();

}

for(i=0;i<n;i++){ d[i].displaya(); d[i].displayb(); d[i].displayc();

}

return 0;

}

**OUTPUT:**

Enter information of students:

For roll number1,

Enter name: akash

Enter Grade: A

For roll number2,

Enter name: adithya

Enter Grade: B

For roll number3,

Enter name: pallavi

Enter Grade: A

Displaying Information:

Roll number: 1

Name: akash

Grade: A

Roll number: 2

Name: adithya

Grade: B

Roll number: 3

Name: pallavi

Grade: A

7. Count the number of persons inside a bank, by increasing count whenever a person enters a bank, using an increment(++) operator overloading function, and decrease the count whenever a person leaves the bank using a decrement(--) operator overloading function inside a class.

**Code** :

#include<iostream> using namespace std;

class counter

{

public:

int count;

counter()

{

count=0;

}

void operator++()

{

count++;

}

void operator--()

{

count--;

}

int display()

{

return count;

}

};

int main()

{

counter c1;

cout<<"\ninitial number of people => "<<c1.display();

++c1;

++c1;

++c1; ++c1;

cout<<"\npresent number => "<<c1.display();

--c1;

cout<<"\nsome of them leave => "<<c1.display();

}

**OUTPUT:**

initial number of people => 0

present number => 4

some of them leave => 3

8. Suppose you have a Piggie Bank with an initial amount of Rs.500 and you have to add some more amount to it. Create a class 'AddAmount' with a data member named 'amount' with an initial value of Rs.500. Now make two constructors of this class as follows:

i - without any parameter - no amount will be added to the Piggie Bank iki - having a parameter which is the amount that will be added to the Piggie Bank Create an object of the 'AddAmount' class and display the final amount in the Piggie Bank.

**Code**:

#include<iostream> using namespace std; int sav=0;

class addamount

{

public: int amount=500; addamount()

{

}

addamount(int a)

{ sav=a;

sav+=amount;

}

void display()

{

cout<<"\nyour savings in a piggy bank : "<<sav;

}

};

int main()

{

int amt;

cout<<"\nenter the amount to add in a piggy bank : "; cin>>amt; addamount s,d(amt); s.display();

return 0;

}

# **OUTPUT:**

enter the amount to add in a piggy bank : 100

your savings in a piggy bank : 600

9. Write a program to print the below pattern? 1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

**Code :**

#include <iostream> using namespace std;

int main()

{ int r, c = 1;

cout << "Enter number of rows: ";

cin >> r;

for(int i = 0; i < r; i++)

{

for(int j= r; j >= i; j--) cout <<" ";

for(int k = 0; k <= i; k++)

{

if (k == 0 || i == 0) c = 1; else c = c\*(i-k+1)/k; cout << c << " ";

}

cout << endl;

}

return 0;

}

**OUTPUT:**

Enter number of rows: 4

1

1 1

1 2 1

1 3 3 1

10. Write a program to print the first n perfect numbers. (Hint Perfect number means a positive integer that is equal to the sum of its proper divisors) Sample Input:

N = 3

Sample Output:

First 3 perfect numbers are: 6 , 28 , 496 Test Cases:

i. N = 0 ii. N = 5 iii. N = -2 iv. N = -5

v. N = 0.2 **Code :**

#include<iostream> using namespace std; int main()

{

int n,i,j,k,c,n1=0; cout<<"\nenter the limit : "; cin>>n;

cout<<"\nperfect numbers : \n"; for(i=1;i<10000;i++){

c=0;

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

if(i%j==0){ c+=j;

}

}

if(i==c){ cout<<i<<"\n";

n1++;

if(n1==n) break;

}

}

return 0;

}

**OUTPUT:**

**enter the limit : 3**

**perfect numbers :**

**6**

**28**

**496**

11. Write a program to print all the composite numbers between a and b?

Sample Input:

1. = 12
2. = 19

Sample Output 14, 15, 16, 18 Test cases:

i. A = 11, B = 11 ii. A = 20, B = 10 iii. A = 0, B = 0 iv. A = -5, B = 5

v. A = 7, B = -12

**Code** :

#include<iostream> using namespace std; int main()

{

int start,end,i,j; cout<<"\nENTER THE STARTING => ";

cin>>start;

cout<<"\nENTER THE ENDING => ";

cin>>end;

for(i=start+1;i<end;i++)

{

int count=0; for(j=1;j<=i;j++)

{

if(i%j==0) count++;

}

if(count==2) continue;

else

{

cout<<i<<"\n";

}

}

}

OUTPUT:

ENTER THE STARTING => 4

ENTER THE ENDING => 9

6

8

12. Write a program to read the numbers until -1 is encountered. Find the average of positive numbers and negative numbers entered by user.

Sample Input:

Enter -1 to exit…

Enter the number: 7

Enter the number: -2

Enter the number: 9

Enter the number: -8

Enter the number: -6

Enter the number: -4

Enter the number: 10

Enter the number: -1 Sample Output:

The average of negative numbers is: -5.0

The average of positive numbers is : 8.66666667 Test cases:

i. -1,43, -87, -29, 1, -9 ii. 73, 7-6,2,10,28,-1 iii. -5, -9, -46,2,5,0 iv. 9, 11, -5, 6, 0,-1 v. -1,-1,-1,-1,-1

**Code:**

#include<iostream> using namespace std; int main()

{

float n,neg=0,pos=0,n1=0,p1=0; cout<<"\nenter -1 to exit !!!"; while(n!=-1){ cout<<"\nenter the num : ";

cin>>n;

if(n<-1){

neg+=n;

n1++;

}

else if(n>0){

pos+=n; p1++;

}

}

cout<<"\npositive average : "<<pos/p1; cout<<"\nnegative average : "<<neg/n1;

return 0;

}

**OUTPUT:**

**enter -1 to exit !!!**

**enter the num : 5**

**enter the num : 7**

**enter the num : 3**

**enter the num : -5**

**enter the num : -3**

**enter the num : -9**

**enter the num : -1**

**positive average : 5**

**negative average : -5.66667**

13. Find the Mean, Median, Mode of the array of numbers? (DAY-3) Sample Input;:

Array of elements = {16, 18, 27, 16, 23, 21, 19} Sample Output:

Mean = 20

Median = 19

Mode = 16 Test cases:

1. Array of elements = {26, 28, 37, 26, 33, 31, 29}
2. Array of elements = {1.6, 1.8, 2.7, 1.6, 2.3, 2.1, .19}
3. Array of elements = {0, 160, 180, 270, 160, 230, 210, 190, 0}
4. Array of elements = {200, 180, 180, 270, 160, 270, 270, 190, 200}
5. Array of elements = {100, 100, 100, 100, 100, 100, 100, 100, 100}

Code:

#include<iostream> using namespace std; int main()

{

int a[50],c=0,i,n,c1,n1,j,k,b[10],re=0,r,max; float mean=0,median=0;

cout<<"\nenter the no of elements in an array : ";

cin>>n;

for(i=0;i<n;i++){ cout<<"\nenter the element "<<i+1<<" : ";

cin>>a[i]; mean+=a[i];

}

for(i=0;i<n;i++){ for(j=i+1;j<n;j++){ k=a[i];

a[i]=a[j];

a[j]=k;

}

}

cout<<"\nmean : "<<mean/n; if(n%2!=0){

n1=n/2;

cout<<"\nmedian : "<<a[n1+1]<<"\n";

}

else{

n1=n/2;

median=(a[n1]+a[n1-1])/2; cout<<"\nmedian : "<<median;

}

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

max=0;

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

if(a[i]==a[j])

max++;

}

if(max>=1){ if(c<max){

c=max;

c1=a[i];

}

}

}

cout<<"\nmode : "<<c1; return 0;

}

**OUTPUT:**

enter the no of elements in an array : 4

enter the element 1 : 23

enter the element 2 : 5

enter the element 3 : 7

enter the element 4 : 7

mean : 10.5

median : 6

mode : 7

14. Find the number of factors for the given number (DAY-3) Sample Input:

Given number: 100 Sample Output:

Number of factors = 9 Test cases:

i. 343 ii. 1080

1. -243
2. 101010
3. 0

**Code :**

#include<iostream> using namespace std; int main()

{

int n,i,j,k; cout<<"\nenter the number : ";

cin>>n;

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

if(n%i==0){

k++;

}

}

cout<<"\nnumber of factors : "<<k; return 0;

}

**OUTPUT:**

enter the number : 12

number of factors : 6

15. Write a program to check if a given year is leap year or not. If it is leap year then print the next leap year, if it is non-leap year then print the previous leap year.

Sample Input:

Enter Date : 1947 Sample Output:

Given year is Non Leap Year

Previous Leap Year: 1944 Test cases:

i. 19.47 ii. 1936

1. 0
2. 2000 v. -1428

**Code**:

#include<iostream>

using namespace std;

int main() {

int year;

cout<<"Enter year : ";

cin>>year;

if(year==0)

cout<<"invalid input!!";

else if(year % 400 == 0)

cout << year << " is a Leap Year";

else if(year % 4 == 0 && year % 100 != 0)

cout << year << " is a Leap Year";

else

cout << year << " is not a Leap Year";

return 0;

}

**OUTPUT:**

Enter year : 2008

2008 is a Leap Year