

ASSIGNMENT 1

- 1.) WAP to enter data of 3 employees and display them using class and object.
- ↳ #include <iostream>
using namespace std;

```
class Employee {
public:
```

```
    string name;
    int age;
    int salary;
```

```
Employee (string s, int age, int sal) {
```

```
    this->name = s;
```

```
    this->age = age;
```

```
    this->salary = sal;
```

```
}
```

```
void print () {
```

```
    cout << this->name << endl;
```

```
    cout << "Age : " << this->age << endl;
```

```
    cout << "Salary : " << this->salary << endl;
```

```
}
```

```
};
```

```
int main () {
```

```
    Employee e1 ("Ankur", 20, 1000000);
```

```
    e1.print();
```

Teacher's Signature _____

Employee e2 ("Raj", 25, 26000);
e2.print();

Employee e3 ("Nerd", 22, 30000);
e3.print();

}

2) Write a program to enter 2 nos. & perform following operations : addition, subtraction, multiplication, division.

↪ #include <iostream>
using namespace std;

class Maths {

public :

int a;

int b;

int sum (int a, int b) {

this → a = a;

this → a = b;

return (this → a + this → b);

}

int multiply (int a, int b) {

this → a = a;

this → b = b;

return (this → a * this → b);

}

```
int minus ( int a, int b ) {
```

```
    this->a = a;
```

```
    this->b = b;
```

```
    return ( this->a - this->b );
```

```
}
```

```
float div ( int a, int b ) {
```

```
    this->a = a;
```

```
    this->b = b;
```

```
    return ( float ) this->a / this->b );
```

```
}
```

```
};
```

```
int main () {
```

```
    int a, b;
```

```
    cin >> a >> b;
```

```
    Maths m;
```

```
    cout << "sum : " << m.sum ( a, b ) << endl;
```

```
    cout << "difference : " << m.minus ( a, b ) << endl;
```

```
    cout << "multiply : " << m.multiply ( a, b ) << endl;
```

```
    cout << "divide : " << m.divide ( a, b ) << endl;
```

```
}
```

- 3) WAP in C++ to input an array having elements . Find sum , largest and smallest element.

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int a[100];
```

```
class Array {
```

```
public :
```

```
int n;
```

```
Array (int n) {  
    this->n = n;  
}
```

```
int sum () {  
    return (accumulate (a, a + this->n, 0));  
}
```

```
int max () {  
    return (*max_element (a, a + this->n));  
}
```

```
int min () {  
    return (*min_element (a, a + this->n));  
}
```

```
};
```

```
int main () {
```

```
int n;
```

```
cin >> n;
```

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

```
    cin >> a[i];
```

```
}
```

```
Array a1 (n);
```

```
cout << "Sum: " << a1.sum() << endl;
```

```
cout << "Max: " << a1.max() << endl;
```

```
cout << "Min: " << a1.min() << endl;
```

```
}
```

Q.) WAP that inputs a character from keyboard & display its corresponding ASCII value.

#include <iostream>

using namespace std;

class Character {

public :

char a ;

Character (char a) {

this → a = a ;

}

void showASCII () {

cout << (int) a << endl ;

}

};

int main () {

char ch ;

cin >> ch ;

Character s (ch) ;

s . showASCII () ;

}

5) WAP using class called Temperature and member functions for a temperature in Fahrenheit and display it in Celsius.

↳ #include <iostream>

using namespace std;

class Temperature {

public :

int temp ;

float ans ;

void showCelsius (int temp) {

this->temp = temp ;

ans = (temp - 32) * (0.55) ;

cout << "TEMP in Celsius : " << ans << endl ;

}

};

int main () {

int t ;

cout << "Input TEMP in Fahrenheit " << endl ;

cin >> t ;

Temperature T ;

T. showCelsius (t) ;

}

ASSIGNMENT 2

1.) WAP in C++ program function using reference variables as arguments to swap the values of a pair of integers.

↳ `#include <iostream>`
`using namespace std;`

```
void swap (int &a, int &b) {  
    int temp;  
    temp = a;  
    a = b;  
    b = temp;  
}
```

```
int main () {  
    int a, b;  
    cout << "Input a & b:" << endl;  
    cin >> a >> b;  
    cout << "a = " << a << " " << "b = " << b << endl;  
    swap (a, b);  
    cout << "After swap : " << endl;  
    cout << "a = " << a << "b = " << b << endl;  
}
```

Q.) WAP in C++ to display Names, Roll NO. & grades of 3 students who have appeared in the examination.

Declare the class of name, Roll No., and grade.

Create an array of class objects. Read & display the contents of the array.

↪ `#include <bits/stdc++.h>`
using namespace std;

class Student {

public :

string name ;

int roll ;

char grade ;

void setter (string name, int roll, char grade)

{

 this->name = name ;

 this->roll = roll ;

 this->grade = grade ;

}

void print ()

{

 cout << "NAME : " << this->name << " " ;

 cout << "ROLL : " << this->roll << " " ;

 cout << "GRADE : " << this->grade << endl ;

}

};

```

int main() {
    Student v[3];
    string name;
    int roll;
    char grade;
    for (int i=0; i<3; ++i) {
        cin >> name >> roll >> grade;
        v[i].setter(name, roll, grade);
        v[i].print();
    }
    return 0;
}

```

- 3.) WAP in C++ to input elements into an integer array & perform following operation :
- display elements of array
 - Sum of elements of array
 - Max. & Min. Elements of array .

```

#include <bits/stdc++.h>
using namespace std;
int arr[100];

```

```

class Array {
public:
    int n;
}

```

```

Array (int n) {
    this->n = n;
}

void display () {
    for (int i = 0; i < this->n; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
}

void sum () {
    cout << "sum: " << accumulate (arr, arr + this->n, 0)
        << endl;
}

void max () {
    cout << "MAX: " << *max_element (arr, arr + this->n)
        << endl;
}

void min () {
    cout << "MIN: " << *min_element (arr, arr + this->n)
        << endl;
}

int main () {
    int n; cin >> n;
    for (int j = 0; j < n; ++j) {
        cin >> arr[j];
    }

    Array a1(n);
    a1.sum(); a1.max(); a1.min();
    return 0;
}

```

Abhishek
29/08/2023

ASSIGNMENT 3

1.) WAP in C++ to find the larger of two given integers. However if the two integers have the same remainder when divided by 7, then return the smaller integer. If the two integers are the same, return 0.

→ #include <bits/stdc++.h>
using namespace std;

```
int main() {
    int a, b, g, s;
    cin >> a >> b;
    if (a > b) {
        g = a; s = b;
    } else if (a < b) {
        g = b; s = a;
    } else if (a == b) cout << 0;
    else if (a % 7 == b % 7) cout << s;
    else cout << g;
    return 0;
}
```

2.) WAP in C++ to check a given array of integers of length 1 or more & return true if 10 appears as either first or last element in the given array.

→ `#include <bits/stdc++.h>`
`using namespace std;`

```
int main() {
    int n; cin >> n;
    int a[n];
    for (int i=0; i<n; i++) { cin >> a[i]; }
    if (a[0] == 10 || a[n-1] == 10) cout << "True" : cout << "False";
    return 0;
}
```

3.) WAP in C++ to count letters, spaces, numbers & other characters in an input string.

→ `#include <bits/stdc++.h>`
`using namespace std;`

```
int main() {
    string s;
    getline (cin, s);
    int letters = 0, spaces = 0, numbers = 0, others = 0;
    for (int i=0; i<s.length(); i++) {
        if (s[i] == ' ') { spaces++; }
        else if ((s[i] >= 'A' && s[i] <= 'Z') ||
                 (s[i] >= 'a' && s[i] <= 'z')) { letters++; }
        else if (s[i] >= '0' && s[i] <= '9') { numbers++; }
        else { others++; }
    }
}
```

```

cout << "letters = " << letters << endl;
cout << "spaces = " << spaces << endl;
cout << "numbers = " << numbers << endl;
cout << "others = " << others << endl;
return 0;
}

```

- 4.) Write a program in C++ to display the following output using a single cout statement.

maths = 90
 physics = 89
 chemistry = 97

→ #include <iostream>
using namespace std;

int main() {

```

cout << "maths = 90" << '\n' << "physics = 89" << '\n'
      << "chemistry = 97" << endl;

```

return 0;

}

5) WAP in C++ to print following output using for loops.

```

      2   2
      3   3   3
      4   4   4   4
      5   5   5   5   5
  
```

↪ #include <bits/stdc++.h>
using namespace std;

```

int main() {
    int n; cin >> n;
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n - i; j++) {
            cout << " ";
        }
        for (int k = 1; k <= 2 * i - 1; k++) {
            if ((k % 2) == 0) {
                cout << i;
            } else {
                cout << ".";
            }
        }
        cout << endl;
    }
    return 0;
}
  
```

ASSIGNMENT 4

1.) WAP in C++ to calculate total salary of employee.

```
#include <iostream>
```

```
using namespace std;
```

```
int main () {
```

```
    cout << "Input salary of every single month : " << endl;
```

```
    int a[12];
```

```
    int sum = 0;
```

```
    for (int i=0; i<12; ++i) {
```

```
        cin >> a[i]; sum += a[i];
```

```
}
```

```
    cout << "Total Salary = " << sum << endl;
```

```
    return 0;
```

```
}
```

2.) WAP in C++ to calculate % of marks of 6 subjects .

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int main () {
```

```
    vector<int> v(6);
```

```
    int sum = 0;
```

```
    cout << "Enter marks of 6 subjects : " << endl;
```

```
    for (int i=0; i<6; ++i) {
```

```
        cin >> v[i]; sum += v[i]; }
```

```
    float per = ((float) sum / 600) * 100;
```

```
    cout << "Percentage = " << per << "%" << endl;
```

```
}
```

3) WAP in C++ to swap 2 nos. without using 3rd variable.

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int main() { void swap (int &a, int &b) {  
    a = a ^ b;  
    b = a ^ b;  
    a = a ^ b;  
}
```

```
int main() {
```

```
    int a, b;
```

```
    cout << "Enter 2 numbers : " << endl;
```

```
    cin >> a >> b;
```

```
    cout << "Number before swap : ";
```

```
    cout << a << " " << b << endl;
```

```
    swap(a,b);
```

```
    cout << "Numbers after swap : ";
```

```
    cout << a << " " << b << endl;
```

```
    return 0;
```

```
}
```

4.) WAP in C++ to find no. of vowels, consonants, digits, spaces in a string.

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int main() {
```

```
    string s;
```

```
    getline (cin, s);
```

Teacher's Signature _____

```

int vowels = 0, con = 0, dig = 0, sp = 0;
for (int i=0; i < s.length(); ++i) {
    if (s[i] == 'a' || s[i] == 'e' || s[i] == 'o' || s[i] == 'i' || s[i] == 'u')
        vowels++;
    else if (s[i] >= '0' && s[i] <= '9') dig++;
    else if (s[i] == ' ') sp++;
    else con++;
}
cout << "Vowels = " << vowels << endl;
cout << "Consonants = " << con << endl;
cout << "Digits = " << dig << endl;
cout << "Spaces = " << sp << endl;
return 0;
}

```

5) WAP in C++ to print all repeated nos. with freq. in array.

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int main() {
```

```
    int n; cin >> n; int a[n]; map<int, int> mp;
```

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

```
        cin >> a[i]; mp[a[i]]++;
```

```
}
```

```
    for (auto v: mp) {
```

```
        if (v.second > 1) {
```

```
            cout << "Num = " << v.first << " ";
```

```
            cout << "Freq = " << v.second << endl; }
```

```
    return 0;
```

ASSIGNMENT 5

- 1.) WAP in C++ to find cube of a number using function.
 → #include <bits/stdc++.h>
 Using namespace std;

```
int cube (int n) {
    return (n*n*n);
}

int main() {
    int n;
    cout << "Enter the no. : ";
    cin >> n;
    cout << "Cube = ";
    cout << cube(n) << endl;
    return 0;
}
```

- 2.) WAP in C++ to do add, subtract & multiplication of two numbers using function.

→ #include <bits/stdc++.h>
 Using namespace std;

```
int add (int a, int b) {
    return (a+b);
}
```

```
int sub (int a, int b) {
    return (a-b);
}
```

```
int mul (int a, int b) {
    return (a*b);
}
```

```
int main () {
    int a, b;
    cout << "Enter nos. ";
    cin >> a >> b;
    cout << "Add = " << add(a,b) << endl;
    cout << "Subtraction = " << sub(a,b) << endl;
    cout << "Multiplication = " << mul (a,b) << endl;
    return 0;
}
```

3) WAP in C++ to check number is palindrome or not using function.

→ ~~#include <bits/stdc++.h>~~

~~using namespace std;~~

```
bool check (string s) {
    int i=0;
    int j = s.size() - 1;
    while (i <= j) {
        if (s[i] == s[j]) {
            return true;
        }
        i++;
        j--;
    }
    return false;
}
```

```

int main() {
    int n;
    cout << "Enter the no. : ";
    cin >> n;
    string s = to_string(n);
    if (check(s)) {
        cout << "YES" << endl;
    }
    else {
        cout << "NO" << endl;
    }
}

```

Q.) WAP in C++ to convert binary no. to decimal no.

→ ~~#include <bits/stdc++.h>~~

~~using namespace std;~~

```

int BtoN(string s) {
    int ans = 0;
    reverse(s.begin(), s.end());
    for (int i=0; i<s.size(); i++) {
        ans += (s[i] - '0') * (1<<i);
    }
    return ans;
}

```

```

int main () {
    int n;
    cout << "Enter the binary no. ";
    cin >> n;
    string s;
    s = to_string (n);
    cout << BtoN(s) << endl;
    return 0;
}

```

5.) WAP in C++ to find factorial of number using function.

```

→ #include <bits/stdc++.h>
using namespace std;
int dp[10000];
int fact (int n) {
    if (n == 0 || n == 1) return 1;
    if (dp[n] != -1) return dp[n];
    return dp[n] = n * fact (n-1);
}

```

```

int main () {
    int n;
    memset (dp, -1, sizeof(dp));
    cout << "Enter no. "; cin >> n;
    cout << "Factorial = " << fact ;
    return 0;
}

```

ASSIGNMENT 6

1) WAP in C++ to delete an element of an array at any position using class.

→ `#include <bits/stdc++.h>`
 using namespace std;

class operation {

public:

void del (int arr[], int n, int pos) {
 for (int i = pos; i < n - 1; i++) {
 arr[i] = arr[i + 1];
 }

cout << "Updated array : " << endl;
 for (int i = 0; i < n - 1; i++) {
 cout << arr[i] << " ";
 }

cout << endl;

}

};

int main () {

int n; cin >> n;

int arr[n];

for (int i = 0; i < n; ++i) {
 cin >> arr[i];

}

int pos; cin >> pos; operation op;
 op.del (arr, n, pos);

}

Teacher's Signature _____

2.) WAP in C++ to count vowels, consonant , digits & special char.
 → #include <iostream>
 using namespace std;

```
class MyClass {
public:
    void myCount (string s) {
        int c = 0, v = 0, d = 0, sp = 0;
        for (int i = 0; i < s.length(); ++i) {
            if (s[i] == 'a' || s[i] == 'e' || s[i] == 'o' ||
                s[i] == 'u') v++;
            else if (s[i] >= 'A' && s[i] <= 'Z') c++;
            else if (s[i] >= '0' && s[i] <= '9') d++;
            else sp++;
        }
        cout << "Vowels : " << v << endl;
        cout << "Consonants : " << c << endl;
        cout << "Digits : " << d << endl;
        cout << "Special characters : " << sp << endl;
    }
}
```

```
int main() {  
    string s; cin >> s;  
    MyClass ct;  
    ct.myCount(s);  
    return 0;  
}
```

3) WAP in C++ to convert Temperature from Celsius to Fahrenheit using class.

→ `#include <bits/stdc++.h>`
using namespace std;

```
class Temp {  
public:  
    int temp;
```

```
void getInPut() {  
    cout << "Enter Temp in °C ";  
    cin >> temp;  
}
```

```
void Cel-To-Feh () {  
    float ans;  
    ans = ((temp) * 9) / 5 + 32;  
    cout << "Fahrenheit Temp : "  
    cout << ans << endl;  
}
```

```

int main () {
    Temp t;
    t. getInPut ();
    t. Cel -> Feh ();
}

```

- Q.) WAP in C++ to check if nos. in array are in AP.
- #include <bits/stdc++.h>
- using namespace std;

```

class MyClass {
public :
    void ap (vector<int> a, int n) {
        vector<int> diff;
        bool b = false;
        for (int i=1; i<n; i++) {
            int x = a[i] - a[i+1];
            diff.push_back (x);
        }
        for (int i=0; i<diff.size () - 1; i++) {
            if (diff[i] == diff[i+1]) {
                cout << "NOT IN AP !!! " << endl;
                b = true;
                break;
            }
        }
        if (b == false)
            cout << "IN AP" << endl;
    }
};

```

```

int main () {
    int n; cin >> n;
    vector<int> a(n);
    for (int j = 0; j < n; j++) {
        cin >> a[j];
    }
}

```

~~My Class check;~~
~~check.ap(a, n);~~
~~return 0;~~

5) WAP in C++ to print product of all odd & even numbers in a range using class.

→ #include <bits/stdc++.h>

using namespace std;

```

class MyClass {
public:
    void product (int a, int b) {
        int odd = 1, even = 1;
        for (int i = a; i <= b; i++) {
            if (i & 1) odd *= i;
            else even *= i;
        }
    }
}

```

```

cout << "ODD PRODUCT = " << odd << endl;
cout << "EVEN PRODUCT = " << even << endl;
}

```

```
int main () {  
    int n; cin >>n;
```

```
My Class cal;  
cal.product ('n');  
return 0;  
}
```

Abhishek
Misra
26/10/23

ASSIGNMENT 7

1.) WAP in C++ to input time in seconds and then convert to hours, minutes and seconds using parametrized constructor.

4) `#include <bits/stdc++.h>`
using namespace std;

```
class MyClass {
public:
    int sec;
    MyClass (int t) {
        sec = t;
        int h = t / 3600;
        int m = (t - h * 3600) / 60;
        int s = t - (h * 3600) - (m * 60);
        cout << h << "hrs " << " ";
        cout << m << "min " << " ";
        cout << s << "sec ";
    }
};

int main () {
    int t;
    cout << " INPUT TIME IN SEC ----> " << endl;
    cin >> t;
    MyClass obj(t);
    return 0;
}
```

2.7 WAP to demonstrate Copy Constructor.

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
class Student {
```

```
    int rno;
```

```
    char name[50];
```

```
    double fee;
```

```
public:
```

```
    Student (int, char [], double);
```

```
    Student (Student &t) {
```

// copy constructor

```
        rno = t.rno;
```

```
        strcpy (name, t.name);
```

```
}
```

```
    void display();
```

```
    void disp () {
```

```
        cout << endl << rno << " " << name;
```

```
}
```

```
} ;
```

```
Student :: Student (int no, char n[], double f) {
```

```
    rno = no;
```

```
    strcpy (name, n);
```

```
    fee = f;
```

```
}
```

```
void Student :: display () {  
    cout << endl << mno << " " << name << " " << fee;  
}
```

```
int main () {  
    Student s (1001, "Manjeet", 10000);  
    s.display ();
```

```
Student manjeet (s);  
manjeet . disp();  
return 0;
```

{

3.) WAP in C++ to create a class with method/function to print area and perimeter of triangle using parametrised constructor.

→ #include <bits/stdc++.h>

using namespace std;

```
class MyClass {  
public:  
    int a, b, c;  
    MyClass (int a, int b, int c) {  
        this->a = a;  
        this->b = b;  
        this->c = c;  
    }
```

```
void funct ();  
};
```

```
void MyClass :: funct() {
    int s = (a+b+c)/2;
    cout << "Perimeter : " << a+b+c << endl;
    double area = sqrt(s)*sqrt(s-a)*sqrt(s-b)*
                  sqrt(s-c);
    cout << "Area : " << area;
}

int main() {
    MyClass triangle(6,6,6);
    triangle.funct();
    return 0;
}
```

ASSIGNMENT 8

- 1.) WAP to demonstrate inline functions and classes.

#include <iostream>

using namespace std;

```
class MyClass {
public:
    inline void funct();
```

}

```
void MyClass :: funct() {
    cout << "This is inline function";
```

}

```
int main() {
```

MyClass obj;

obj.funct();

return 0;

}

- 2.) WAP to convert Temperature from Celsius to Fahrenheit and vice-versa using inline function.

#include <iostream>

using namespace std;

```
class MyClass {
```

public:

inline void CtoF();

inline void FtoC();

}

```

void MyClass :: CtoF() {
    cout << "Enter Temp in deg C" << endl;
    float c; cin >> c;
    cout << (float)((c * 9) / 5) + 32 << endl;
}

void MyClass :: FtoC() {
    cout << "Enter Temp in deg F" << endl;
    float f; cin >> f;
    cout << (float)((f - 32) * 5) / 9 << endl;
}

int main() {
    MyClass obj1; obj1.CtoF(); obj1.FtoC();
}

```

3) WAP to compute sum of a variable number of integers passed as arguments.

```

#include <iostream.h>

using namespace std;
class MyClass {
public:
    void sum (int a, ...) {
        va_list va;
        int sum = 0;
        va_start(va, a);
        for (int i = 0; i < a; i++) {
            sum += va_arg(va, int);
        }
        cout << sum;
    }
};

```

```
int main () {  
    MyClass obj1;  
    obj1.sum (2, 13, 15);  
    return 0;  
}
```

4) WAP to check if a given function is even or odd using inline function.

→ #include <iostream>

```
using namespace std;
```

```
class MyClass {
```

```
public:
```

```
    inline void check (int n) {
```

```
        if (n % 2) cout << "ODD" << endl;
```

```
        else cout << "EVEN" << endl;
```

```
    }};
```

```
int main () {
```

```
    MyClass obj1;
```

```
    int n; cin >> n;
```

```
    obj1.check (n); return 0;
```

```
}
```

5) WAP to print pattern:

a) #include <iostream>

```
using namespace std;
```

```
void print1 () {
```

```
    int n; cin >> n;
```

```
    for (int i=n, j=1; i>=1; i--) {
```

```
        for (int j=1; j<=n-i; j++) { cout << " "; }
```

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

```
        cout << endl;
```

```
}
```

Teacher's Signature _____

```

void print2() {
    int n; cin >> n;
    for (int i=1; i<=2*n-1; i++) {
        for (int j=1; j<=2*n-1; j++) {
            int x = abs(n-i);
            int y = abs(n-j);
            int max-val = x > y ? x : y;
            cout << max-val + 1;
        }
        cout << endl;
    }
}

```

```

void print3() {
    int n; cin >> n;
    for (int i=1; i<=n/2; i++) {
        for (int j=1; j<=i-1; j++) { cout << " ";}
        char ch = 'A';
        for (int j=1; j<=n-i+1; j++) {
            cout << ch;
            ch++;
        }
        cout << endl;
    }
}

```

```

void print4() {
    int n; cin >> n;
    for (int i=1; i<=n; i++) {
        for (int j=1; j<=n-i; j++) { cout << " ";}
        for (int j=1; j<=2*i-1; j++) {
            if (j == 1 || j == 2*i-1) {
                if (j == 1)
                    cout << 1;
            } else {
                cout << 2;
            }
        }
    }
}

```

```

else {
    cout << " ";
}

cout << endl;

for (int i=n; i>1; i--) {
    for (int j=0; j<=n-i; j++) {
        cout << " ";
    }

    for (int j=1; j<=2*i-3; j++) {
        if (j==1 || j==2*i-3) {
            if (j==1)
                cout << 1;
        }
        else {
            cout << 2;
        }
    }

    cout << endl;
}

int main() { printf(); print&(); print3(); print4();
    return 0;
}

```

Teacher's Signature

~~Abhishek~~
311023

ASSIGNMENT 9

- 1) Type a class employee as shown & create an array of objects called as manager[2], workers[3] & display.

→ `#include <iostream>`

`using namespace std;`

`class Employee {`

`char name[30]; int age;`

`public:`

`void putdata() {`

`cout << "Enter name" << endl; cin >> name;`

`cout << "Enter age" << endl; cin >> age; }`

`void getdata() {`

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

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

`int main() {`

`Employee manager[2], workers[3];`

~~`for (int i=0; i<2; i++) { manager[i].putdata(); }`~~

~~`for (int i=0; i<3; i++) { workers[i].putdata(); }`~~

~~`for (int i=0; i<2; i++) { manager[i].getdata(); }`~~

~~`for (int i=0; i<3; i++) { workers[i].getdata(); }`~~

~~`return 0; }`~~

- 2) Write a C++ program to add two numbers using object as function argument (use call by value method).

→ `#include <iostream>`

`using namespace std;`

`class MyClass {`

`public:`

`int x, y;`

```

myClass ( int x, int y ) { this->x = x; this->y = y; }
void sum ( myClass obj ) { cout << obj.x + obj.y; }

int main() {
    myClass obj (10, 20);
    obj.sum (obj); return 0;
}

```

3) WAP to swap two values using object as function argument.
→ #include <iostream.h>

```

using namespace std;
class myClass { public : int x, y;
    myClass ( int x, int y ) { this->x = x, this->y = y; }
    void swapper ( myClass &obj ) {
        int temp = obj.x; obj.x = obj.y; obj.y = temp;
        cout << obj.x << " " << obj.y; }
}

int main () { int x, y; cin >> x >> y; myClass obj(x, y); obj.swapper(obj);
}

```

4) WAP to add x & y using friend function.
→ #include <iostream.h>

```

using namespace std;
class myClass { int x, y;
public :
    myClass ( int x, int y ) { this->x = x; this->y = y; }
    friend void sum ( myClass ); }

void sum ( myClass obj ) {
    cout << obj.x + obj.y; }

int main () {
    int x, y; cin >> x >> y;
    myClass obj (x, y);
    sum (obj);
    return 0;
}

```

ASSIGNMENT TO

1) WAP for Inheritance beyond single level.

```
#include <iostream>
```

```
using namespace std;
```

```
class A { public: A() { cout << "Hello in Base Class\n"; } };
```

```
class B { public: B() { cout << "Second Base Class\n"; } };
```

```
class C : public A, public B {
```

```
public: C() { cout << "Derived Class.\n"; } };
```

```
int main() { C obj; }
```

2) WAP to enter patient details using inheritance.

```
#include <iostream>
```

```
using namespace std;
```

```
class Hospital { public: string name; int age; int id; } ;
```

```
class Patient : public Hospital {
```

```
public:
```

```
Patient() { cin >> name; cin >> age; cin >> id; }
```

```
void display() { cout << name; cout << age; cout << id; } };
```

```
int main() { Patient P; P.display(); }
```

3) WAP to demonstrate an example of single inheritance..

```
#include <iostream>
```

```
using namespace std;
```

```
class A { public: A() { cout << "Base Class\n"; } };
```

```
class B : public A {
```

```
public:
```

```
B() { cout << "Inherited class in single level inheritance\n"; }
```

```
};
```

```
int main() {
```

```
    B obj;
```

```
}
```

4) WAP to demonstrate example of multiple inheritance.

```
#include <iostream>
using namespace std;
class A { public: A() { cout << "Base Class 1\n"; } };
class B { public: B() { cout << "Second Base Class 1\n"; } };
class C : public A, public B {
    public: C() { cout << "Derived in Multiple Inheritance\n"; } ;
} int main() { C obj; }
```

5) WAP to demonstrate example of hybrid inheritance.

```
#include <iostream>
using namespace std;
class Vehicle () { public: Vehicle () { cout << "Vehicle 1\n"; } };
class Fare () { public: Fare () { cout << "Fare 1\n"; } };
class Car : public Vehicle {};
class Bus : public Vehicle, public Fare {};
int main () { Bus obj; return 0; }
```

6) WAP to show access to private, public & protected using inheritance.

```
#include <iostream>
using namespace std;
class Base { private: int pvt = 1;
protected: int prot = 2;
public: int pub = 3;
int getPVT () { return pvt; } };

```

```
class PublicDerived : public Base {
    public: int getProt () { return prot; } };
int main () {
```

PublicDerived obj;

```
cout << obj.getPVT () << "\n" << obj.getProt () << "\n" <<
obj.pub;
```

}.

7) WAP to override the member function using inheritance.

```
#include <iostream>
using namespace std;
class Base { public : void print() { cout << "Base Function " << "\n"; } };
class Derived : public Base {
    public : void print() { cout << "Derived Function " << "\n"; }
} int main() {
    Derived derived; derived.print(); return 0;
}
```

8) WAP to demonstrate an example of multilevel inheritance.

```
#include <iostream>
using namespace std;
class MyClass { public : MyClass () { cout << "Parent Class In "; } };
class MyChild : public MyClass {
    public : MyChild () { cout << "Child Class In "; }
} class MyGrandChild : public MyChild {
    public : MyGrandChild () { cout << "Grandchild class In "; }
} int main() {
    MyGrandChild myObj;
}
```

9) WAP to illustrate use of constructors in single inheritance.

→ Already done in ques 1.

~~Answer sheet~~