**NOTES**

Oops (object oriented programing)

-encapsulation

-class

-object

-abstraction

-inheritance

-polymorphism

Three types of A&

-public

-private

-protected

DIFFERENCES

FEATURE ENCAPUSLATION ABSTRACTION

-FOUCS HOW DATA IS HIDDEN INSIDE WHAT FUNCTIONALITIES

A CLASS ARE EXPOSED

ACHIEVED BY ACESS MODIFIERS(PRIVATE,PUBLIC)

Inheritance means to change one class to another class

#include <iostream>

**using namespace std;**

**class car{**

**public://public acess specifier**

**string brand;**

**string model;**

**int price;**

**int year;**

**//function to display the output**

**void display()**

**{**

**cout<<"brand name"<<endl;**

**cout<<"model name"<<endl;**

**cout<<"price:"<<price<<endl;**

**cout<<"year of manfacturing"<<year<<endl;**

**}**

**};**

**int main(){**

**car car1;//creating an object of the class car**

**cout<<"enter the brand name :"<<endl;**

**//cin >>car1.brand:**

**getline(cin,car1.brand);**

**cout<<"enter the model name :"<<endl;**

**getline (cin ,car1.model);**

**cout<<"enter the price :"<<endl;**

**cin >>car1.price;**

**cout<<"enter the Man.F year :"<<endl;**

**cin >>car1.year;**

**car1.brand="BMW";**

**car1.model="B\_class";**

**car1.price=5000000;**

**car1.year=2024;**

**cout<<car1.brand<<endl;//direct way to printing**

**cout<<car1.model<<endl;//direct way to printing**

**cout<<car1.price<<endl;//direct way to printing**

**cout<<car1.year<<endl;//direct way to printing**

**car1.display();//disply the o/p using function**

2.program

#include <iostream>

using namespace std;

class teacher{

public://public acess specifier

string name;

string subject;

char gender;

int year;

//function to display the output

void display()

{

cout<<"name"<<endl;

cout<<"subject"<<endl;

cout<<"gender:"<<endl;

cout<<"year "<<year<<endl;

}

};

int main() {

teacher T1;

cout<<"enter the teacher\_name :"<<endl;

cin >> T1.name;

cout<<"enter the teacher\_subject :"<<endl;

cin>> T1.subject ;

cout<<"enter the gender :"<<endl;

cin >> T1.gender;

cout<<"enter the year :"<<endl;

cin >> T1.year;

T1.display();

return 0;

}

3.program

#include <iostream>

using namespace std;

class Teacher

{

private:

//data memebers

string name;

string subject;

int years;

public:

void setdetail(string tname,string tsubject,int tyears)

{

name=tname;

subject=tsubject;

years=tyears;

}

//member functions

void display()

{

cout<<"name:"<<name

<<",subject:"<<subject

<<",year:"<<years<<endl;

}

};

int main()

{

//create 2 object

Teacher t1,t2,t3;

//assign value to object

/\*t1.name = "Riya";

t1.subject = "Electronics";

t1.year = 9;\*/

t1.setdetail("Riya","Electronics",9);

t2.setdetail("Raj","cs",4);

t3.setdetail("Rakesh","cs",6);

t1.display();

t2.display();

t3.display();

return 0;

}

name:Riya,subject:Electronics,year:9

name:Raj,subject:cs,year:4

name:Rakesh,subject:cs,year:6

=== Code Execution Successful ===

**3.program**

// Online C++ compiler to run C++ program online

#include <iostream>

using namespace std;

class car{

public:

string make;

string model;

int year;

//defalut constructor (no parameters)

car(){

make="BENZ";

model="XO8";

year=2;

}

void displayinfo(){

cout<<year<<""<<make<<" "<<model<<endl;

}

};

int main(){

car mycar;

mycar.displayinfo();

return 0;

}

2BENZ XO8

=== Code Execution Successful ===

2.type

#include <iostream>

using namespace std;

class Teacher {

public:

string name;

string dept;

int experience;

double salary;

//constructor

Teacher()

{

name="Arib";

dept="computer science";

experience=10;

salary=75000;

cout<<"Teacher name:"<<name<<endl;

cout<<"depertment:"<<dept<<endl;

cout<<"experience:"<<experience<<endl;

cout<<"salary:"<<salary<<endl;

}

};

int main()

{

Teacher T1;//create an object

return 0;

}

Teacher name:Arib

depertment:computer science

experience:10

salary:75000

=== Code Execution Successful ===

3.type

#include <iostream>

using namespace std;

class Teacher {

public:

string name;

string dept;

int experience;

double salary;

//constructor

Teacher()

{

cout<<"Teacher name:"<<"Arib"<<endl;

cout<<"depertment:"<<"computer science"<<endl;

cout<<"experience:"<<10<<endl;

cout<<"salary:"<<75000<<endl;

}

};

int main()

{

Teacher T1;//create an object

return 0;

}

Teacher name:Arib

depertment:computer science

experience:10

salary:75000

=== Code Execution Successful ===

4.type

#include <iostream>

using namespace std;

class Teacher {

public:

string name;

string dept;

int experience;

int salary;

//constructor

Teacher(string n,string d,int e,int s)

{

cout<<"Teacher name:"<<n<<endl;

cout<<"depertment:"<<d<<endl;

cout<<"experience:"<<e<<endl;

cout<<"salary:"<<s<<endl;

}

};

int main()

{

Teacher T1("Arib","computer science",10,75000);//create an object

cout<<endl;

Teacher T2("zeeshan","c++",5,800000);

}

Teacher name:Arib

depertment:computer science

experience:10

salary:75000

Teacher name:zeeshan

depertment:c++

experience:5

salary:800000

=== Code Execution Successful ===

3.program

#include <iostream>

using namespace std;

class parent

{

public:

void food()

{

cout<<"Inherited burger";

}

};

class child : public parent//.creating the child class and inherting it

{};

int main()

{

child b;

b.food();

return 0;

}

Inherited burger

=== Code Execution Successful ===