MODULE #3 INTRODUCTION TO OOPS PROGRAMMING

1.What are the key differences between Procedural Programming and Object-Oriented Programming (OOP)?

Key diffrences : code organization , data hiding , code reusability , complexity management , error handling.

2. List and explain the main advantages of OOP over POP.

Advantage of oop :

- Flexibility : oop flexible because code is define at run time.
- Reusability : oop reuse code through inheritance.
- security : oop use encapsulation to hide data from user.
- Maintainability : oop easy maintain and upgrade because rich libraries.

Advantage of pop :

- Easy to understand : pop follow top todown approach flow.
- No object or classes : pop program without creating class and object.

3. Explain the steps involved in setting up a C++ development environment.

1.Google search the c++ compiler or IDE DEV C++.

2.choice IDE:

Dev c++ : download and install the dev c++ valid link in browser and click download button.

3.configure in IDE :

Dev c++ : install the dev c++ after the create new file and write code.

4.save a program and run :

Dev c++ : first program execute in dev c++. .cpp extension file save and run program.

4. What are the main input/output operations in C++? Provide examples.

Input : user from the value cin(>>) left shift operator.

Output : user print the value cout(<<)right shift operator.

Example :

```
#include<iostream>

Using namespace std;

Main()

{

Int a;

Cout<<”enter the value”;

Cin>>a;

Cout<<”\n”<<a;

}
```

1.What are the different data types available in C++? Explain with examples.

Data type: data type is type of data and store variable and value.

Datatype is two type : 1.primitive datatype.

2.nonprimitive datatype.

- **Primitive datatype :** primitive datatype is provide ianguage. Ex : int , float , char etc…
- **nonprimitive datatype :** non primitive datatype is provide developer. Ex string , array , structure etc…
- **Integer :** integer store the positive value small range. integer size 4 bytes.
- **Float :** float store the decimal value small range. Float size 4 bytes.
- **Long integer :** store the positive value high range. Long integer size 4 bytes.
- **double :** store the decimal value high range. Double size 8 bytes.
- **character:** character store the single character value. Character size 1 bytes
- **String :** string declare store the group of character value.
- **Boolean** : Boolean value declare true or false. Boolean size 1 bytes.

2. Explain the difference between implicit and explicit type conversion in C++.

Implicit: implicit type conversion is compiler automatic convert type casting.

Example:

```
int a=10;

float b=12.12,c;


c=a+b;

cout<<"\nc:"<<c;
```

Explicit: explicit type conversion is user convert type casting.

Example:

```
int x=10;
```

```
float y=12.12,z;

z=x+(int)y;

cout<<"\nz:"<<z;
```

3. What are the different types of operatorsin C++? Provide examples of each.

Operator : operator is symbol and perform the operation from value. Multiple operator in c++ language

1.arithmatic operator : perform the mathematical operation use arithmetic operator.

Ex : +(addition) , -(substarction) , *(multiplication), /(division) , %(parectege).

```
int a=10,b=20;

cout<<"\naddition:"<<a+b;

cout<<"\nsubtraction:"<<a-b;

cout<<"\nmultiplication:"<<a*b;

cout<<"\ndivision:"<<a/b;

cout<<"\nmodual:"<<a%b;
```

2. relational operator: perform the two value comparison use relational operator.

Ex : ==(double equal) , !=(not equal) , >(greter then) , <=(lessthen equal) , =>(greterthen equal)

```
int a1=50,b1=100;

cout<<"\nlessthen:"<<(a1<b1);
```

```
cout<<"\ngreterthen:"<<(a1>b1);

cout<<"\nlessthenequal:"<<(a1<=b1);

cout<<"\ngreterthenequal:"<<(a1>=b1);

cout<<"\nequal:"<<(a1==b1);

cout<<"\nnotequal:"<<(a1!=b1);
```

3.logical operator : perform the two or more condition check use logical operator.

Ex : &&(and) , ||(or) , !(not)

```
int x=10,y=10,z=30;
```

AND:

```
if(x==y && y==z)
{
        cout<<"\nsame";
}
else
{
        cout<<"\ndiffrent";
}
```

OR:

```
if(x==y || y==z || z==x)
{
        cout<<"\nsame";
}
```

```
else
{
        cout<<"\ndiffrent";
}
```

NOT:

```
if(!(x==z))
{
        cout<<"\ndiffrent";
}
else
{
        cout<<"\nsame";
}
```

4.assignment operator : perform the assign value use assignment operator.

Ex: += , -+, *= , /=

a+=b , a=a+b

a-=b , a=a-b

a*=b , a=a*b

a/=b , a=a/b

5.increment/decrement : increment operator value+1,decrement operator value-1.

Prefix : ++a , --a

postfix : a++ , a--

bitwise operator :

ex : & , | , << , >>

4. Explain the purpose and use of constants and literals in C++.

Constants : provide value for fix cannot change after the define. Constants variable is use const keyword.

Ex :

```
const int a=20;
Main()
{
Cout<<"\na:"<<a;
}
A=40;  //arror
Cout<<"\na:"<<a;
```

Literals: specify data value directly in the code is literal. Program fix value specify using literals a four type literals integer , float , character , string.

Ex : int a=10;   float a=20.30   char ch='R'   string name="shyam"

1.What are conditional statements in C++? Explain the if-else and switch statements.

1.If stetment : use the if stetment cheak only true condition check .

Syntex :

```
if(condition)
{
//stetment
}
```

2.if_else stetment : use the if_else stetment check the true or false condition.

Syntex :

```
if(condition)
{
//stetment
}
Else
{
//stetment
 }
```

3.switch stetment : multiple choice value and user one choice use switch stetment.

Not use relation operator , switch use int , charcter datatype , switch use keyword switch , break , case , default.

Syntex :

```
switch(choice)
{
Case 1:
// stetment
 Break;
Case 2 :
//stetment
Break;
 Default :
//stetment
}
```

2. What is the difference between for, while, and do-while loops in C++?

- While loop : while(condition) { execute the code } . while loop condition is true loop is execute and condition is false loop is not execute. while loop is entry control loop, entry control loop is check condition before execute code. Exit the loop condition is false.
- For loop : for(initialization , condition , increment/decrement). for loop condition is true loop is execute and condition is false loop is not execute. for loop is entry control loop, entry control loop is check condition before execute code. Exit the loop condition is false.
- Do while loop : do { execute the code } while(condition) . do while loop condition first time execute and after condition is true loop is execute and condition is false loop is not execute. do while loop is exit control loop, exit control loop is execute code before check condition. Exit the loop condition is false.

While loop :

```
I = 1;(initialization) : starting point
While(I <= 5)condition : ending point
{
Cout<< I ; //execute code
I++; : increment/decrement
}
```

For loop :

```
for(I = 1(initialization; I <= 5 (condition)
;i++(increment/decrement))

{

Cout<< I ; // execute code

}
```

Do while loop :

```
I = 1;(initialization)

Do

{

Cout<< I ; // execute code

I++; : increment/decrement

} While (I <= 5); : (condition)
```

3. How are break and continue statements used in loops? Provide examples

Break : break is keyword and break keyword use the break the code , rest of the code will not executed.

//break :

```
#include<iostream>

Using namespace std;

Main()

{
```

```
int i;
for(i=0;i<=10;i++)
{
If(i==6);
Break;
Cout<<"\n"<<i;
}}
```

continue : continue is keyword and continue keyword use the skip the code , rest of the code will be executed.

//continue :

```
#include<iostream>
Using namespace std;
Main()
{
int i;
for(i=0;i<=10;i++)
{
If(i==6);
Break;
Cout<<"\n"<<i;
}}
```

4. Explain nested control structures with an example.

nested control structure : nested control structure refer the control structure within another control structure.

Syntex :

```
if(condition
{
If(condition)
{
//stetment
}
else
{
 //stetment
}
else
{
//stetment
}
```

1. **What is a function in C++? Explain the concept of function declaration, definition, and calling.**

   Function is block of code in c++. use the function () round breaket .

   function is two type :

   1) Build in function : build in function is already c++ language provide.it is not created by developer.

   2) User define function : user define function is provide by developer.

   - **Mainly four type of user define function:**

   1) without argument and without return value
   2) without argument and with return value

3) with argument and without return type
4) with argument and with return type

- **Function mainly three part :**

  o Function Declaration

  o Function call

  o Function definition

Example:

```
#include<iostream>

Using namespace std;

 Void num();

Void num()

{

Int I;

For(i=0;i<=10;i++)

{

Cout<<"\n"<<I;

}

}

Main()

{

Num();

}
```

2. What is the scope of variables in C++? Differentiate between local and global scope.

scope of variables: scope of variable is area of the variable use the variable valid area in c++.

Two type of scope of variable:

1.globle variable: globle variable is define outside the function and valid scope function outside and inside.

2.local variable: local variable is define inside the function and valid scope function inside. Not valid scop outside otherwise error ganrrate.

Example:

```
#include<iostream>
using namespace std;
int gv=10;
void abc();
void abc()
{
int lv=20;
cout<<"\ninside globle variable"<<gv;
cout<<"\ninside local variable"<<lv;
}
main()
{
cout<<"\noutside globle variable"<<gv;
//cout<<"\ninside local variable"<<lv;
abc();   }
```

3.Explain recursion in C++ with an example.

MODULE #3 INTRODUCTION TO OOPS PROGRAMMING

Recursive : recursive is function define and function call it self.

Recursive is part of function in c++.

Example:

```
#include<iostream
Using namespace std;
Void factorial(int);
Void factorial(int n)
{
If(n==0)
{
Return 1;
}
Else
{
Return fact*factorial(n-1);
}
}
Main()
{
Int n;
Cout<<”enter the n:”;
Cin>>n;
Int ans=factorial(n);
Cout<<”\n”<<ans;
}
```

4.What are function prototypes in C++? Why are they used?

Function prototype used: function prototype used in c++ inform the compiler a function detail a function use , return type , parameter , name .

Function prototype:

Function declaration

Function calling

Function definition

Function main

Example:

```
#include<iosream>
Using namespace std;
Int add(int,int);    //declare function
Int add(int x, int y)   //define function
{
Return x+y;
}
Main()    // main function
{
Int x,y;
Cout<<"enter the x and y";
Cin>>x>>y;
Int ans=add(x,y);   //call function

Cout<<ans;

}
```

1.What are arrays in C++? Explain the difference between single-dimensional and multidimensional arrays.

Array : array is collaction of element a different data type.

Array is three type :

1) one dimensional array
2) two dimension array
3) multi dimension array

Differentiate between one-dimensional and multi-dimensional arrays:

One dimensional : it has only one dimensional. Array store single line multiple element. One dimensional array is execute series type.

Multi dimensional : it has three or multiple dimensional. Multi dimensional array store row and column multiple element. And multi dimensional array is execute table or matrix.

2.Explain string handling in C++ with examples.

String handling use function and handle the string.

strlen() : find out the length of the string

Example :

```
String str[20];

cout<<"enter the string";
```

```
cin>>str;
cout<<strlen(str));
```

strcpy() : to copy one string to another.

Example :

```
String str1[20], str2[20];
Cout<<“enter the string1”;
Cin>>str1;
Cout<<strcpy(str2, str1));
```

strcat() : To concate two strings.

Example :

```
string str1[20], str2[20], str3[20];
cout<<”enter the string1”;
cin>>str1;
cout<<“enter the string2”;
cin>>str2;
strcpy(str3, strcat(str1, str2));
cout<<str3;
```

strcmp() : To compare two strings. (by default refer the case).

Example :

```
String str1[20],str2[20];
if(strcmp(str1, str2)==0)
        cout<<"\n\n\t Strings are equal.";
else
        cout<<"\n\n\t Strings are not equal.";
```

strchr() : return a pointer position of the first charchter in string.

Example :

```
String str[]="hello world";
char *ptr=strchr(str,'w');
if(ptr!=NULL)
{
        printf("%s",ptr);
}
```

3.How are arrays initialized in C++? Provide examples of both 1D and 2D arrays.

Array is three type :

1) one dimensional array
2) two dimension array
3) multi dimensional array

one dimensional initialized : int arr[5]={8,6,4,2,9};

example :

```
int arr[5]={10,20,30,40,50};
int i;
for(i=0;i<5;i++)
{
        Cout<<arr[i];   }
```

two dimensional initialized : int arr[2][2]={[8,9],[3,5],[1,2]}

example:

```
#include<iostream>
Using namespace std;
main()
{
        int arr[2][2]={{10,20},{30,40}};
        int r,c;
        for(r=0;r<2;r++)
        {
        for(c=0;c<2;c++)
                {
                        Cout<<arr[r][c];
                }
                Cout<< “\n";
}
}
```

3. Explain string operations and functions in C++.

string operations:

- **gets()** : to read the string with space.
- **puts()** : to print the string
- **strlen()** : to find out the length of the string
- **strrev()** : o reverse the string
- **strlwr()** : to convert into lower case
- **strupr()** : to convert into upper case
- **strcpy()** : to copy one string to another.

- **strcat()** : To concate two strings.
- **strcmp()** : To compare two strings. (by default refer the case)
- **stricmp()** : To compare two strings. (by ignoring the case)

1.Explain the key concepts of Object-Oriented Programming (OOP).

Key concept of oop :

1.class , 2.object , 3.inheritance , 4.polymorphisum , 5.encapculation , 6.abstraction .

1.class : a class is blueprint or template that define data member and member function .

Example : class class name.

2.object : object is instant of the class. A basic unit of oop. Object is user assigned memory data member and member function.

Example : class name object name;

Object name . function name();

3.Inheritancre : inheritance is one class inherit properties to another class.

Base class , perent class , super class inherit child class , drived class , sub class.

And inheritance multiple class call one object.

Inheritance five type : simple inheritance , multiple inheritance , multilevel inheritance , hyerarchical inheritance , hybrid inheritance

4.**encapculation :** wrapping up a data store one unit. A data and method hiding and improve the security of the code.

5.abstrction : abstraction is most important concept of object oriented programming .

Abstraction is hiding any background information from the outside world.

It is used implement the class to provide data security.

6.polymorphisum : one name having many form.

Polymorphisum two type :

1) Compile time polymorphisum (static binding) - Overloading

    -Function(method) Overloading

    -Operator Overloading

2) Run time polymorphisum (Dynamic binding) - Overriding

    -Function(method) Overriding

2. What are classes and objects in C++? Provide an example.

a class is blueprint or template that define data member and member function .

object is instant of the class. A basic unit of oop. Object is user assigned memory data member and member function.

Example :

```
#include<iostream>

Using namespace std;

Class data

{

Private:

        Int age;

        String name , hobby;

Public:

        Void get();

        Void print();

};

Void data::ger()

{

        Cout<<”enter the age”;

        Cin>>age;

        Cout<<”enter the name”;

        Cin>>name;

        Cout<<”enter the hobby”;

        Cin>>hobby;

}

Void data::print()
```

```
{
        Cout<<"\nage:"<<age;
        Cout<<"\nname:"<<name;
        Cout<<"\nhobby:"<<hobby;
}
Main()
{
        Data obj;
        Obj . get();
        Obj . print();
}
```

3.What is inheritance in C++? Explain with an example.

Inheritancre : inheritance is one class inherit properties to anther class.

Base class , perent class , super class inherit child class , drived class , sub class.

And inheritance multiple class call one object.

Inheritance five type :

- **simple inheritance**
- **multiple inheritance**
- **multilevel inheritance**
- **hyerarchical inheritance**
- **hybrid inheritance**

Example :

```
#include<iostream>
```

```
Using namespace std;
Class abc
{
Protected:
        Int rollno;
        String name;
Public:
        Void get();
};
Void abc::get()
{
        Cout<<"enter the rollno";
        Cin>>rollno;
        Cout<<"enter the name";
        Cin>>name;
}
Class xyz : public abc
{
Protected:
        String cours;
Public:
        Void get1();
        Void print();
}
```

```
Void xyz::get1()

{

        Cout<<"enter the course";

        Cin>>course;

}

Void xyz::print()

{

        Cout<<"\nrollno:"<<rollno;

        Cout<<"\nname:"<<name;

        Cout<<"\ncourse:"<<course;

}

Main()

{

        Xyz obj;

        Obj.get();

        Obj.get1();

        Obj.print();

}
```

4.What is encapsulation in C++? How is it achieved in classes?

encapsulation : wrapping up a data store one unit. A data and method hiding and improve the security of the code.

Encapsulation data member and member function store access modifier(specifier)

Public , private , protected.

Public : use data member and member function inside the class and outside the class.

Private : use data member and member function inside the class.

Protected : use data member and member function to class to another class means one class proparties use another class