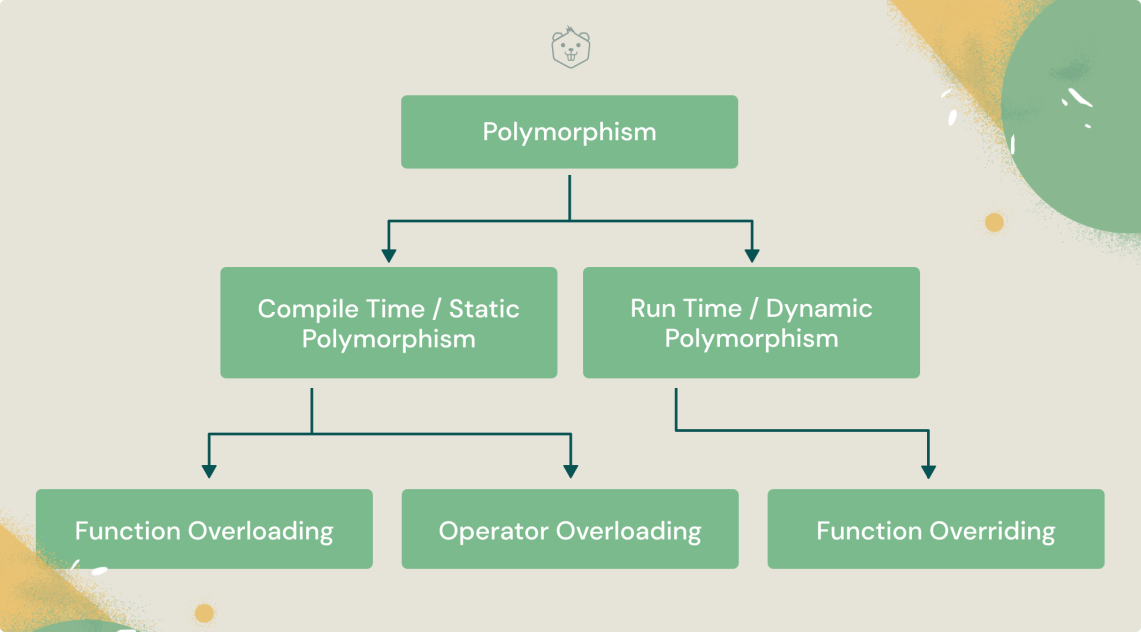
**Practical 8**

**Aim:** Implementing the concept of Polymorphism in CPP

## Theory:

**Polymorphism**

The word “polymorphism” means having many forms. In simple words, we can define polymorphism as the ability of a message to be displayed in more than one form. A real-life example of polymorphism is a person who at the same time can have different characteristics. A man at the same time is a father, a husband, and an employee. So the same person exhibits different behavior in different situations. This is called polymorphism. Polymorphism is considered one of the important features of Object-Oriented Programming.



**Polymorphism is of 2 types:**

1. **Compile time**: ****Compile-time polymorphism**** means the binding is occurring at compile time. It can be achieved through static binding.
2. **Run time**: ****Run-time polymorphism**** is where at run time we came to know which method is going to invoke. It can be achieved through dynamic binding.

We can implement polymorphism in C++ using the following ways:

1. [Function overloading](https://www.programiz.com/cpp-programming/function-overloading)
2. [Operator overloading](https://www.programiz.com/cpp-programming/operator-overloading)
3. [Function overriding](https://www.programiz.com/cpp-programming/function-overriding)
4. [Virtual functions](https://www.programiz.com/cpp-programming/virtual-functions)

## C++ Function Overriding:

* In [C++ inheritance](https://www.programiz.com/cpp-programming/inheritance), we can have the same function in the base class as well as its derived classes.
* When we call the function using an object of the derived class, the function of the derived class is executed instead of the one in the base class.
* So, different functions are executed depending on the object calling the function.

This is known as ****function overriding**** in C++.

## C++ Operator Overloading:

* In C++, we can overload an operator as long as we are operating on user-defined types like objects or structures.
* We cannot use operator overloading for basic types such as int, double, etc.
* Operator overloading is basically function overloading, where different operator functions have the same symbol but different operands. And, depending on the operands, different operator functions are executed.

**C++ Function Overriding**

* In [C++ inheritance](https://www.programiz.com/cpp-programming/inheritance), we can have the same function in the base class as well as its derived classes.
* When we call the function using an object of the derived class, the function of the derived class is executed instead of the one in the base class.
* So, different functions are executed depending on the object calling the function.

This is known as ****function overriding**** in C++.

**Practical Related Questions:**

1. What is polymorphism?
2. **What are the types of Polymorphism?**
3. [What is function overloading in C++?](https://examradar.com/oop-using-cpp-cpp-programming-polymorphism-based-questions-answers/" \l "8_What_is_function_overloading_in_C" \o "8. What is function overloading in C++?)

**Programs:**

1. Overload the operator so that two strings can be concatenated.
2. Write a program to implement function overloading for finding area of circle and rectangle.
3. Write a program for the following class hierarchy for the employee where the base class is employee and derived class programmer and manager:.Here make display function virtual which is common for all and which will display information of programmer and manager interactively.

**Conclusion :-**

Hence, we learnt to implement the concept of Function Overloading, Function Overriding, operator overloading in CPP.

**PRACTICAL 8**

**PROBLEM 1:** **Overload ‘+’ unary operator so that two strings can be concatenated.**

#include<iostream.h>

#include<string.h>

Class String Demo{

Char str1[20];

public:

void get(){

cin>>str1;

}

void put(){

cout<<”\n”<<str1;

}

StringDemo operator + (StringDemo SS){

StringDemo temp;

strcpy(temp.str1,SS.str1);

strcat(str1,temp.str1);

strcpy(temp.str1,str1);

return temp;

}

};

void main(){

StringDemo sd1,sd2,sd3;

clrscr();

cout<<”\nEnter first string:”;

sd1.get();

cout<<”\nEnter second string:”;

sd2.get();

sd1.put();

sd2.put()

sd3=sd1+sd2;

cout<<”\nConcatinated string:”;

sd3.put();

getch();

}

**PROBLEM 2: Write a program to implement function overloadoing for Area of Circle and Rectangle.**

#include<iostream.h>

#include<conio.h>

float area(float r);

float(int,int);

float area(float r)

{

return 3.14\*r\*r;

}

float area(int l, int b)

{

return l\*b;

}

void main()

{

float Result,r;

int result,l,b;

cout<<"Enter radius:"<<endl;

cin>>r;

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

cout<<'Enter length and breadth:"<<endl;

cin>>l>>b;

cout<<"Area of Rectangle:"<<area(l,b)<<endl;

getch();

}

**PROBLEM 3: Write a program for following class hiearchy for employee where the base class is an employee and derived classes are programmer and manager.Here make this display function as virtual function which is common for all which display information of programmer and manager interactively.**

#include<iosteeam.h>

#include<conio.h>

class Employee

{

public:

virtual void display()

{

cout<<"Employee class";

}

};

class Programmer : public Employee

{

public:

void display()

{

cout<<"\n Programmer class";

}

};

class Manager : public Employee

{

public:

void display()

{

cout<<"\n Manager class";

}

};

void main()

{

Employee e,\*ptr;

Programmer p;

Manager m;

ptr=&e;

ptr-> display();

ptr=&p;

ptr-> display();

ptr=&m;

ptr-> display();

getch();

}