EXPERIMENT - 02

TITLE: Function Overloading

AIM: Write a C++ program that illustrates the concept of Function over loading.

OBJECTIVES: To study the concept of polymorphism and function overloading.

THEORY:

Polymorphism: Polymorphism is another important OOP concept. An operation may exhibit different behavior in different instances. Using a single function name to perform different type of task is known as function overloading.

Overloading: Overloading occurs when the same operator or function name is used with different signatures (in different form)

• for e.g., +/*/- used to do different behavior than usual.

• Both operators and functions can be overloaded in C++

Function overloading: Function overloading is a type of polymorphism that allows multiple functions to share the same name with different parameters. The compiler identifies the function either on the basis of the number of parameters, the data type of the parameters or the order of the data type of the parameters passed to the function.

Different ways to Overload a Function

1. By changing number of Arguments: In this type of function overloading, we define two functions with same names but different number of parameters of the same type.

2. By having different types of argument: In this type of overloading, we define two or more functions with same name and same number of parameters, but the type of parameter is different.

Advantages of function overloading:

- The main advantage of function overloading is to improve the code readability and allows code reusability.
- The same apparent function can be called to perform similar but different tasks.

CODE:

```
# include<iostream>
using namespace std;
int sum (int a,int b)
{
return a+b;
}
int sum ( int a, int b, int c)
{
return a+b+c;
}
int sum (int a, int b, int c, int d)
{
return a+b+c+d;
}
int main()
{
int choice;
char ans='y';
do
{
cout<<"\n main menu";
cout<<"\n1 addition of 10 and 20 nos";
cout<<"\n2 addition of 10,20 and 30 nos";
cout<<"\n3 addition of 10,20,30 and 40 nos";
cout<<"\n4 exit";
cout<<"\n enter your choice:";
cin>>choice;
switch(choice)
{
case 1:cout<<" result="<<sum(10,20);
break;
case 2:cout<<"result="<<sum(10,20,30);
break;
case 3:cout<<"result="<<sum(10,20,30,40);
break;
default: cout<<"exiting";
}
cout<<"\n do you want to go back to main menu?";
cin>>ans;
}
while (ans='y' || ans == 'Y');
return 0;
}
```

CONCLUSION: