

Function Overloading

Ali Haider

syedalihaider.ciit@gmail.com

Department of Computer Science IUB

Overview

- Function Overloading
- Constructor Overloading
- Examples
- Homework Tasks

Function Overloading

- Function overloading is a feature in C++ where two or more functions can have the same name but different parameters.
- Function overloading can be considered as an example of polymorphism feature in C++.
- **For Example**
- `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; }`

Cont...

- `int main()`
- `{`
- `cout<<"sum of 10 and 20 is "<<sum(10,20)<<endl;`
- `cout<<"sum of 10, 20 and 30 is "<<sum(10,20,30)<<endl;`
- `cout<<"sum of 10, 20, 30 and 40 is "<<sum(10,20,30,40)<<endl;`
- `return 0;`
- `}`

Cont...

- `float findAvg(float a,float b){return (a+b)/2;}`
- `float findAvg(float a,float b,float c){return (a+b+c)/3;}`
- `int main()`
- `{`
- `cout<<"AVG of 10.3 and 20.33 is "<<findAvg(10.3,20.33)<<endl;`
- `cout<<"sum of 10, 20 and 30 is "<<findAvg(10,20,30)<<endl;`
- `return 0;`
- `}`

Example

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

void print(int i) {
    cout << " Here is int " << i << endl;
}

void print(double f) {
    cout << " Here is float " << f << endl;
}

void print(char const *c) {
    cout << " Here is char* " << c << endl;
}

int main() {
    print(10);
    print(10.10);
    print("ten");
    return 0;
}
```

C:\Users\ShahTab\Documents\OOP\functionOverloading1.exe

```
Here is int 10
Here is float 10.1
Here is char* ten
```

```
-----
Process exited after 0.1925 seconds with return value 0
Press any key to continue . . .
```

Constructor Overloading

- In C++, We can have more than one constructor in a class with same name, as long as each has a different list of arguments.
- This concept is known as Constructor Overloading and is quite similar to function overloading.
- Overloaded constructors essentially have the same name (name of the class) and different number of arguments.
- A constructor is called depending upon the number and type of arguments passed.
- While creating the object, arguments must be passed to let compiler know, which constructor needs to be called.

Example

```
#include <iostream>
#include<string>
using namespace std;
class Student{
    private:
        string std_Name;
        int    std_Age;
        int    std_Marks;
    public:
        Student():std_Name("Ali"),std_Age(25),std_Marks(80)
        {
        }
        Student(string n,int a,int m):std_Name(n),std_Age(a),std_Marks(m)
        {
        }
        void show(){
            cout<<"Name is : "<<std_Name<<endl;
            cout<<"Age is : "<<std_Age<<endl;
            cout<<"Marks are : "<<std_Marks<<endl;
        }
};
int main(){
    Student s;
    Student s1("Ikram",27,70);
    s.show();
    s1.show();

    return 0;
}
```

C:\Users\ShahTab\Documents\OOP\constructorOverloading.exe

Name is : Ali
Age is : 25
Marks are : 80
Name is : Ikram
Age is : 27
Marks are : 70

Process exited after 0.269 seconds with return value 0
Press any key to continue . . .

Homework Tasks

- Write a c++ program to overload product() function. Create four product functions first function take 2 parameter of integer type, second function take 3 parameters of integer type, third function take 4 parameters of integer type and fourth function take 5 parameters of integer type. Use all these created functions in main() function. Hint:
`int product(int a,int b){ return a*b;}`
- Write a c++ code for creating Class Employee (emp_Name,emp_Dept,emp_Salary) with two constructor with default value and assigned at declaration time and create a function to show data of employee. Create object of Employee class in main function.