

Aim: To study functions in C++ programming**Theory:**

Reference Variables to Functions: In C++, you can create references to functions just like you can create references to variables. A reference to a function allows you to call the function using a different name, and it's often used for cleaner syntax or to create aliases for functions.

Inline Functions: An inline function is a function that the compiler treats as if the code were inserted directly into the calling code, instead of performing a regular function call. This can lead to performance improvements by reducing the overhead of function calls. To declare an inline function, you typically use the inline keyword.

Friend Functions: A friend function in C++ is a function that is not a member of a class but is allowed to access the private and protected members of that class. It's typically declared within the class, but it's defined outside of the class scope. This is often used to provide external functions with privileged access to class internals.

Function Overloading: Function overloading allows you to define multiple functions with the same name but different parameter lists. The compiler determines which function to call based on the number and types of arguments provided during the function call. It's a form of polymorphism that makes code more readable and versatile.

Expt 2A : Ref Variable : WAP C++ to check if a number is Krishnamurthy number(number whose sum of the factorial of digits is equal to the number itself. For example $145, 1! + 4! + 5! = 1 + (4*3*2*1) + (5*4*4*2*1) = 1 + 24 + 120 = 145$.) Pass the number as a reference variable to function.

```
#include<iostream>

using namespace std;

int fac(int num){
    int result = 1;
    while(num){
        result *= num--;
    }
    return result;
}

int KMN(int &num){
    int tmp = num, result = 0;
    while(tmp){
        result += fac(tmp % 10);
        tmp /= 10;
    }
    return result;
}

int main()
{
    int num;
    cout<<"Enter a Number: "; cin>>num;
    if(num == KMN(num))
        cout<<"Entered Number is a Krishnamuthy
Number"<<endl;
    else
        cout<<"Entered Number is not a Krishnamuthy
Number"<<endl;
    return 0;
}
```

```
PS C:\Users\Asus-PC\Desktop\collage\opps\e
Enter a Number: 145
Entered Number is a Krishnamuthy Number
```

Expt 2B: Inline function : WAP to compute arithmetic operations on 2 numbers by declaring the arithmetic functions as inline.

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

```
inline float addition(float a, float b){return a + b;}
inline float subtraction(float a, float b){return a - b;}
inline float division(float a, float b){return a / b;}
inline float multiplication(float a, float b){return a * b;}
```

```
int main()
{
    float num1, num2;
    int op;
    while (true)
    {
        cout << "Enter operations" << endl;
        cout <<
"1.Addition\n2.Subtraction\n3.Division\n4.Multiplicat
ion\n5.exit" << endl;
        cin >> op;
        if(op == 5) exit(1);
        cout << "Enter first number : ";
        cin >> num1;
        cout << "Enter second number : ";
        cin >> num2;

        switch (op)
        {
            case 1:
                cout << "Sum: "<<addition(num1,
num2)<<endl; break;
            case 2:
                cout<<"Sub: "<<subtraction(num1,
num2)<<endl; break;
            case 3:
                cout<<"Div: "<<division(num1, num2)<<endl;
break;
            case 4:
                cout<<"Multi: "<<multiplication(num1,
num2)<<endl; break;
            case 5: exit(1);
            default : cout<<"wrong input!"<<endl; break;
        }
    }
}
```

```
PS C:\Users\Asus-PC\Desktop\collage\opps\e
PS C:\Users\Asus-PC\Desktop\collage\opps\e
Enter operations
1.Addition
2.Subtraction
3.Division
4.Multiplication
5.exit
1
Enter first number : 12
Enter second number : 1
Sum: 13
Enter operations
1.Addition
2.Subtraction
3.Division
4.Multiplication
5.exit
2
Enter first number : 4
Enter second number : 2
Sub: 2
Enter operations
1.Addition
2.Subtraction
3.Division
4.Multiplication
5.exit
4
Enter first number : 2
Enter second number : 3
Multi: 6
Enter operations
1.Addition
2.Subtraction
3.Division
4.Multiplication
5.exit
3
Enter first number : 1
Enter second number : 2
Div: 0.5
Enter operations
1.Addition
2.Subtraction
3.Division
4.Multiplication
5.exit
5
PS C:\Users\Asus-PC\Desktop\collage\opps\e
```

Expt 2C: Function Overloading : WAP to compute area of diff shapes(circle,rectangle,triangle) using function overloading.

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

double area(double r){
    return (3.14 * r * r);
}
double area(double a, double b){
    return (a * b);
}
double area(double b, double h, int t){
    return (0.5 * b * h);
}

int main()
{
    int op;
    while(true)
    {
        cout<<"\n1.Area of circle\n2.Area of
rectangle\n3.Area of triangle\n4.exit"<<endl;
        cin>>op;

        switch(op)
        {
            case 1:
                double r;
                cout<<"Enter radius of circle: "; cin>>r;
                cout<<"Area: "<<area(r);
                break;
            case 2:
                double x, y;
                cout<<"Enter Both sides of rectangle: ";
cin>>x>>y;
                cout<<"Area: "<<area(x,y);
                break;
            case 3:
                double b, h;
                cout<<"Enter Base and Height of triangle: ";
cin>>b>>h;
                cout<<"Area: "<<area(b,h,1);
                break;
            case 4: exit(1);
            default : cout<<"Wrong Input! Enter again!
"<<endl; break;
        }
    }
}
```

```
PS C:\Users\Asus-PC\Desktop\collage\opps\e
PS C:\Users\Asus-PC\Desktop\collage\opps\e

1.Area of circle
2.Area of rectangle
3.Area of triangle
4.exit
1
Enter radius of circle: 7
Area: 153.86
1.Area of circle
2.Area of rectangle
3.Area of triangle
4.exit
2
Enter Both sides of rectangle: 12
3
Area: 36
1.Area of circle
2.Area of rectangle
3.Area of triangle
4.exit
3
Enter Base and Height of triangle: 5
2
Area: 5
1.Area of circle
2.Area of rectangle
3.Area of triangle
4.exit
4
```

Expt 2D: Friend function : WAP to Swap the two contents of the classes using a friend function

```
#include<iostream>
#include<string.h>

using namespace std;
class B;
class A{
    int age;
    char name[10];
    public:
    A(){
        cout<<"Enter name: "; cin>>name;
        cout<<"Enter age: "; cin>>age;
    }
    void get_data(){cout<<"Name:
"<<name<<endl<<"Age: "<<age<<endl;}
    friend void swap(A& ,A&);
};

void swap(A &a, A &b){
    int age;
    char name[10];
    age = a.age;
    a.age = b.age;
    b.age = age;
    strcpy(name,a.name);
    strcpy(a.name,b.name);
    strcpy(b.name,name);
}

int main(){
    cout<<"Student 1:"<<endl;
    A x;
    cout<<"Student 2:"<<endl;
    A y;
    swap(x,y);
    cout<<"Student 1:"<<endl;
    x.get_data();
    cout<<"Student 2:"<<endl;
    y.get_data();
    return 0;
}
```

```
PS C:\Users\Asus-PC\Desktop\collage\opps\e
PS C:\Users\Asus-PC\Desktop\collage\opps\e

1.Area of circle
2.Area of rectangle
3.Area of triangle
4.exit
1
Enter radius of circle: 7
Area: 153.86
1.Area of circle
2.Area of rectangle
3.Area of triangle
4.exit
2
Enter Both sides of rectangle: 12
3
Area: 36
1.Area of circle
2.Area of rectangle
3.Area of triangle
4.exit
3
Enter Base and Height of triangle: 5
2
Area: 5
1.Area of circle
2.Area of rectangle
3.Area of triangle
4.exit
4
```

Conclusion: The concepts of refrence variable in a function, inline function, friend function and function overloading were understood and implemented.

Nitesh Naik

(Subject Faculty)

