

### LAB 3

- i. WAP to find area of a circle, a rectangle and a triangle, using concept of function overloading.
- ii. WAP to find volume of a sphere, a cylinder and a cuboid, using function overloading.
- iii. WAP which displays a given character, n number of times, using a function. When the n value is not provided, it should print the given character 80 times. When both the character and n value is not provided, it should print '\*' character 80 times.

[Write the above program in two ways:-

-using function overloading.

-using default arguments.]

```
Void print(char c='*', int n=80);
```

```
Void print(char c);
```

```
Void print(int n);
```

```
Void print(char c, int n);
```

```
Void print(void)
```

- iv. WAP to find square and cube of a number using inline function.

```
Inline int sq(int I) { return I*I;}
```

- v. WAP to increment the value of an argument given to function USING INLINE function.

```
Inline int incr(int I) { return ++i; }
```

Vi. Write a program to create a class called COMPLEX and implement the following overloading functions ADD that return a COMPLEX number.

a) ADD (int a , complex s2) - where a is an integer (real part) and s2 is a complex number.

```
Complex add(int a, complex s2)
```

```
{
```

```
Complex t;
```

```
T. real=a+s2.real;
```

```
T.img=s2.img;
```

```
Return t; }
```

b) ADD (s1, s2) - where s1 and s2 are complex numbers.

vii. Write a program to find the summation of three numbers by using one function only with function name SUM having three arguments. If at run time one argument is given to the function SUM, then second and third argument will be assumed by default as 10 and 20 respectively. If two arguments are given at run time, then third argument will be assumed as 20. Use function with default argument concepts.

Viii. Write a program to demonstrate the concept of call-by-value, call-by-reference & call-by address by taking swapping of two numbers as an example.

**Ix.** Write a program to demonstrate the use of scope resolution operator(::) by declaring same variable name globally and locally and display the value of global and local variables.