Functions in C++

By Avani M. Sakhapara

Assistant Professor,

IT dept, KJSCE

Inline function

- In case of normal functions, the compiler have to jump to another location for the execution of the function and then the control is returned back to the instruction immediately after the function call statement.
- So execution time taken is more in case of normal functions and memory space is saved.
- But for small functions, a substantial amount of execution time may be spent in this overhead.

Inline function

- These are the functions designed to speed up program execution.
- An inline function is expanded in the line where it is invoked/called i.e. the compiler replaces the function call with the corresponding function code.
- Usually functions that can be defined in one or two lines are made inline.(small functions)

Inline function

- A function can be made inline by prefixing the keyword **inline** to the function definition.
- syntax:

```
inline function_header
{
body of the function
}
```

Inline function

- Note:

Inline keyword merely sends a request to the compiler. The compiler may ignore this request if the function definition is too long or too complicated and compile the function as a normal function.

Inline function

The inlining does not work for the following situations :

- For functions returning values and having a *loop* or a *switch* or a goto statement.
- For functions that do not return value and having a return statement.
- For functions having static variable(s).
- If the inline functions are recursive (i.e. a function defined in terms of itself).

Function Overloading

- More than one function having the same function name but different number of arguments or different type of arguments.
- It does not depend on the return type of the function.
- Eg: int volume(int s) //cube

  double volume(double r, int h) //cylinder

Function Overloading

- A function call first matches the function definition having the same number and type of arguments. If found then it uses that function.
- If exact match is not found then the compiler uses implicit conversions. If the conversion results in multiple matches then the compiler generates an error message.

Function Overloading

Example:

1. int add (int , int);
2. float add (float, int);

Function calls

a) int s=add(8 , 2);
b) float x=add(8.5 , 2);