

C++ Programming

Ch. 8 Adventures in Functions

Spring 2014

Myung-Il Roh

Department of Naval Architecture and Ocean Engineering
Seoul National University

Ch. 8 Adventures in Functions

Contents

- ☑ C++ Inline Functions
- ☑ Reference Variables
- ☑ Default Arguments
- ☑ Function Overloading
- ☑ Summary
- ☑ Practice

C++ Inline Functions (1/2)

☑ Inline Function

- We can use the inline function to reduce time for jumping between memories during the function call.

☑ The Way to Declare Inline Function

- Preface the function declaration with the keyword `inline`.
 - Ex. `inline double square(double x) { return x*x; }`
- The function definition comes before the first use of the function.

☑ Limitations

- `inline` is not supported for all compilers, but
- Use when the function spends time in a loop or a branching statement.
- An inline facility should be used only when necessary.

C++ Inline Functions (2/2)

- Regular Functions vs. Inline Regular Functions

```
int main()
```

```
{
```

```
...
```

```
hubba(2);
```

```
...
```

```
hubba(4);
```

```
...
```

```
}
```

```
void hubba(int n)
```

```
{
```

```
for (int i = 0; i < n; i++)  
    cout << "hubba" << "\n";
```

```
}
```

- **Regular function**

A regular function

```
int main()
```

```
{
```

```
...
```

```
{
```

```
n = 2;
```

```
for (int i = 0; i < n; i++)
```

```
    cout << "hubba" << "\n";
```

```
}
```

```
...
```

```
{
```

```
n = 4;
```

```
for (int i = 0; i < n; i++)
```

```
    cout << "hubba" << "\n";
```

```
}
```

```
...
```

```
}
```

- **Inline function**

An inline function

Reference Variables

- ☑ 'Reference' is a name that acts as an alias, or an alternative name, for a previously defined variable.

- ☑ Expression
 - Declare a reference variable with ' &' symbol before the variable name.
 - Ex.

```
int rats;                // Declaration of a variable
int &rodents = rats;     // Declaration of a reference variable
```

- ☑ References as Function Parameters
 - When references are used as function parameters, (like 'Passing by Address').

. Thus, the references should be initialized when they are declared.

Default Arguments

- ☑ A default argument is _____ if we omit the corresponding actual argument from a function call.

- ☑ Expression

- Set up default values on the function prototype.
- If a default value is set up on a function parameter, all the next parameters should have a default value. ➡

- Ex.

```
int harpo(int n, int m = 4, int j = 5);    // OK
int harpo(int n, int m = 4, int j);      // Error
```

- Actual arguments from the function call are

- Ex.

```
int harpo(int n, int m = 4, int j = 5);
beeps = harpo(2);                          // Same as '      '.
```

Function Overloading (Function Polymorphism)

: list of the type of function parameters

- Ex. `void print(double d, int width);` ➔ Signature is ' ' .

☑ Function Overloading

- It is to use multiple functions sharing the same name.
- We can use function overloading
- The function overloading can be used in case of the function signature is different (variable names are doesn't matter). That is, we can use the same function name when

- Ex.

```
long gronk(int n, float m);           // Same signature  
double gronk(int n, float m);       //
```

```
long gronk(int n, float m);           // Different signature  
double gronk(float n, float m);      //
```


Summary

- ☑ If we use an inline keyword with a function definition, instead of having the program jump to a separate section of code to execute the function, the compiler

that lets us

create an alias for a variable.

- ☑ If a function call omits the corresponding argument, the program uses the default value.

- ☑ A function's signature is . We can define two functions having the same name, provided that they have different signatures. This is called

Practice 1

- ☑ Make a function that switches 2 variables ('swap function').
 - Use pointers
 - Use references

Practice 2

- ☑ Make a program that calculates the area of a circle.
 - If the function is **called by one argument**, make a function that calculates the area of a circle with a parameter which is the radius.

 - If the function is **called by two argument**, make a function that calculates the area of a fan-shaped region with two parameters one is the radius, the other is the internal angle