CSC404 PROGRAMMING II
FUNCTION PART 1

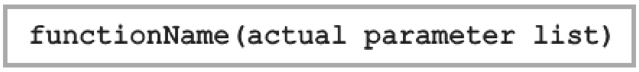
Fadilah Ezlina Shahbudin

Introduction

- Functions are often called <u>modules.</u>
- They are like miniature programs that can be combined to form larger programs.
- They allow complicated programs to be divided into manageable pieces.

Function Call

- Using the function by writing the name of the function.
- Provide the information (arguments) the function.
- Syntax to call a value-returning function:



```
functionName(actual parameter list)
```

Predefined Functions

- Also called library function or built-in function.
- In C++, a function is similar to that of a function in algebra
  - It has a name
  - It does some computation
- Some of the predefined mathematical functions are:

```
sqrt(x)
pow(x, y)
floor(x)
```

Predefined Functions (cont'd.)

- Predefined functions are organized into separate libraries:
  - I/O functions are in `iostream` header
  - Math functions are in `cmath` header
- To use predefined functions, you must include the header file using an `#include` statement.

Header File

Header File	Type of functions
`conio`	Declares variable functions used in calling the operating system console I/O routine
`ctype`	Contains information used by the character classification and character conversion macros (such as isalpha and toascii)
`math`	Declares prototypes for the math functions and math errors handlers
`stdlib`	Declares several commonly used routines such as conversion and search/sort routines
`string`	Declares several string-manipulation and memory-manipulation routines
`time`	Defines the structure filled in by the time-conversion and a type used by the routines
`iomanip`	Declares the C++ streams I/Omanipulators and contains templates for creating parameterized manipulators
`iostream`	Declares the basic C++ streams I/O routines

Trigonometric Functions

Function	Description
`sin(radians)`	Returns the trigonometric sine of an angle in radians.
`cos(radians)`	Returns the trigonometric cosine of an angle in radians.
`tan(radians)`	Returns the trigonometric tangent of an angle in radians.
`asin(a)`	Returns the angle in radians for the inverse of sine.
`acos(a)`	Returns the angle in radians for the inverse of cosine.
`atan(a)`	Returns the angle in radians for the inverse of tangent.

Exponent Functions

Function	Description
`exp(x)`	Returns e raised to power of x (e^x).
`log(x)`	Returns the natural logarithm of x ($\ln(x) = \log_e(x)$).
`log10(x)`	Returns the base 10 logarithm of x ($\log_{10}(x)$).
`pow(a, b)`	Returns a raised to the power of b (a^b).
`sqrt(x)`	Returns the square root of x ($\sqrt{x}$) for x >= 0.

Rounding Functions

Function	Description
`ceil(x)`	x is rounded up to its nearest integer. This integer is returned as a double value.
`floor(x)`	x is rounded down to its nearest integer. This integer is returned as a double value.

The min, max, and abs Functions

- □ The `min` function return the minimum of two numbers.
- □ The `max` function return the maximum of two numbers.
- □ The `abs` function return the absolute value of the number (`int, long, float,` or `double`).

The min, max, and abs Functions

- `max(2, 3)` returns **3**
- `max(2.5, 4.0)` returns **4.0**
- `min(2.5, 4.6)` returns **2.5**
- `abs(-2)` returns **2**
- `abs(-2.1)` returns **2.1**

Character Functions

require **`cctype`** header file

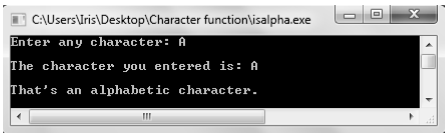
FUNCTION	MEANING
`isalpha`	**true** if arg. is a letter, **false** otherwise
`isalnum`	**true** if arg. is a letter or digit, **false** otherwise
`isdigit`	**true** if arg. is a digit 0-9, **false** otherwise
`islower`	**true** if arg. is lowercase letter, **false** otherwise

Character Functions

require **`cctype`** header file

FUNCTION	MEANING
`isprint`	**true** if arg. is a printable character, **false** otherwise
`ispunct`	**true** if arg. is a punctuation character, **false** otherwise
`isupper`	**true** if arg. is an uppercase letter, **false** otherwise
`isspace`	**true** if arg. is a whitespace character, **false** otherwise

Example: isalpha function



```
Enter any character: A
The character you entered is: A
That's an alphabetic character.
```

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

int main(){
  char input;
  cout << "Enter any character: ";
  cin >> input;

  cout << "\nThe character you entered is: " << input <<
  endl;

  if (isalpha(input))
     cout << "\nThat's an alphabetic character.\n" << endl;

  return 0;
}
```

Power function

- □ `pow(x, y)`, calculates `xy`; that is, the value of `pow(x, y)= xy`.
- □ For example, `pow(2, 3)=` 2^3 `= 8.0` and `pow(2.5, 3)=` 2.5^3 `= 15.625`.
- □ Because the value of `pow(x, y)` is of type `double`, we say that the function `pow` is of type `double` or that the function `pow` returns a value of type `double`.
- □ Moreover, `x` and `y` are called the parameters (or arguments) of the function `pow`.
- □ Function `pow` has two parameters.

Square root function

- □ `sqrt(x)`, calculates the non-negative square root of `x` for `x >= 0.0`. For example, `sqrt(2.25)` is `1.5`.
- □ The function `sqrt` is of type double and has only one parameter.
- □ Example: Find the hypotenuse of a right triangle
- □ Formula:

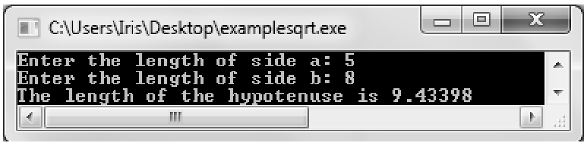
  $c=\sqrt{a^2 + b^2}$

Example: Square root function

```
#include <iostream>
#include <cmath>
using namespace std;
int main(){
   double a, b, c;

   cout << "Enter the length of side a: ";
   cin  >> a;
   cout << "Enter the length of side b: ";
   cin  >> b;

   c = sqrt(pow(a, 2.0) + pow(b, 2.0));
   cout << "The length of the hypotenuse is ";
   cout << c << endl;
   return 0;
}
```

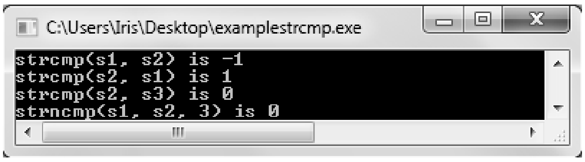


```
Enter the length of side a: 5
Enter the length of side b: 8
The length of the hypotenuse is 9.43398
```

Floor function

- `floor(x)`, calculates the largest whole number that is less than or equal to x.
- For example, `floor(48.79)` is `48.0`.
- The function floor is of type `double` and has only one parameter.

Example: strcmp & strncmp



```
strcmp(s1, s2) is -1
strcmp(s2, s1) is 1
strcmp(s2, s3) is 0
strncmp(s1, s2, 3) is 0
```

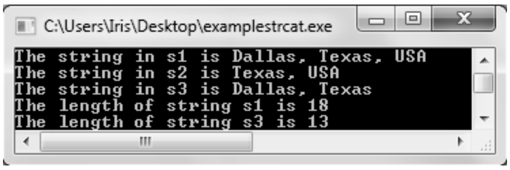
```
#include <iostream>
#include <cstring>
using namespace std;

int main(){
  char *s1 = "abcdefg";
  char *s2 = "abcdg";
  char *s3 = "abcdg";

  cout << "strcmp(s1, s2) is " << strcmp(s1, s2) << endl;
  cout << "strcmp(s2, s1) is " << strcmp(s2, s1) << endl;
  cout << "strcmp(s2, s3) is " << strcmp(s2, s3) << endl;
  cout << "strncmp(s1, s2, 3) is " << strncmp(s1, s2, 3) << endl;

  return 0;
}
```

Example: strcat & strlen



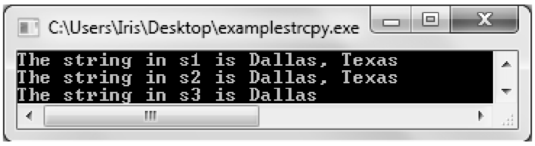
```
The string in s1 is Dallas, Texas, USA
The string in s2 is Texas, USA
The string in s3 is Dallas, Texas
The length of string s1 is 18
The length of string s3 is 13
```

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

int main(){
  char s1[20] = "Dallas";
  char s2[20] = "Texas, USA";
  char s3[20] = "Dallas";
  strcat(strcat(s1, ", "), s2);
  strncat(strcat(s3, ", "), s2, 5);
  cout << "The string in s1 is " << s1 << endl;
  cout << "The string in s2 is " << s2 << endl;
  cout << "The string in s3 is " << s3 << endl;
  cout << "The length of string s1 is " << strlen(s1) << endl;
  cout << "The length of string s3 is " << strlen(s3) << endl;
  return 0;
}
```

Example: strcpy & strncpy

```
#include <iostream>
#include <cstring>
using namespace std;
int main(){
  char s1[20];
  char s2[20] = "Dallas, Texas";
  char s3[20] = "AAAAAAAAAA";
  strcpy(s1, s2);
  strncpy(s3, s2, 6);
  s3[6] = '\0'; // Insert null terminator
  cout << "The string in s1 is " << s1 << endl;
  cout << "The string in s2 is " << s2 << endl;
  cout << "The string in s3 is " << s3 << endl;
  return 0;
}
```



```
The string in s1 is Dallas, Texas
The string in s2 is Dallas, Texas
The string in s3 is Dallas
```