UTM
UNIVERSITI TEKNOLOGI MALAYSIA

String Manipulations Additional Slides

Programming Technique II

(SECSJ1023)

December 2020

String data

- String data is a series of characters.

  Example:

  Hello World

  1 plus two = 3

- Enclosed with double quotes, e.g. : **"ABC"**

- Terminated with a NULL character ('\0'),

  "ABC" is stored as as a series of characters 'A', 'B', 'B','\0'

Example:

```
cout << "UTM Skudai" << endl;    // output: UTM Skudai
cout << "UTM\0 Skudai" << endl; // output: UTM
```

String data (2)

- Each character in a string is stored as an ASCII code

  Example:

  `"ABC"` is represented in memory as `[65,66,67,0]`

  `"abc"` is represented in memory as `[97,98,99,0]`

  Where ASCII code for A is 65, B is 66 and so on, a is 97, b is 98 and so on.

Example:

```
if ("ABC" < "abc"){  // This condition is true. Lower case letters have greater ASCII codes
    cout << "Capital case is indeed less than the lower case" << endl;
}
```

- C++ supports two types of strings:
  - C strings
  - C++ strings

C vs C++ Strings

	C String	C++ String
Data representation	• A C string is stored as an array of characters. • In C, arrays are treated as pointers	• A C++ string is stored as an object • It is treated as regular data (not a pointer)
Declaration	*Example:* `char name[20] = "John";` `char name[] = "John";` `char *name = "John";`	*Example:* `string name="John";` `string name("John");`
Manipulation	• Follows procedural programming • using functions and parameter passing • Need to include **<cstring>** library	• Follows OOP style • Using methods and overloaded operators from the string objects

C vs C++ Strings (2)

	C String	C++ String
Referencing string characters	C string is referenced as regular array *Example:* `char name[20] = "John";` **`cout << name[0];`** *// J* **`cout << name[2];`** *// h*	Referenced as an array or using method *Example:* `string name = "John";` **`cout << name[0];`** *// J* **`cout << name[2];`** *// h* **`cout << name.at(0);`** *// J*
Assignment	*Example:* `char name[20] = "John";` *// To change the name* **`strcpy`**`(name,"Johnny");` *// Cannot use the assignment operator* `name = "Johnny";`	*Example:* `string name = "John";` *// Using method* **`name.assign("Johnny");`** *// Using the assignment operator* **`name = "Johnny";`**

C vs C++ Strings (3)

	C String	C++ String
Comparison	• **Do not use relational operators.** • **It means, comparing between pointers (not the strings)** *Example:* `char name1[20] = "John";` `char name2[20] = "John";` `char name3[20] = "Zack";` `if (name1 == name2){ …. }` *// The condition is **false**. name1 and name2* *// point to different location* `if (name3 < name1){ …. }` *// The condition is **true**. Memory allocation follows* *// stack structure. name3 is on top of name1*	**C++ string is treated as regular data (not a pointer)** *Example:* `string name1 = "John";` `string name2 = "John";` `string name3 = "Zack";` `if (name1 == name2){ …. }` *// The condition is **true**. Both strings* *// hold the same data* `if (name3 < name1){ …. }` *// The condition is **false**. Z is larger than J*

C vs C++ Strings (4)

	C String	C++ String
Comparison (cont.)	**To compare C strings, use function strcmp()** Note this function returns an integer (not a Boolean). 0 : both strings are exactly the same < 0 : the left string is smaller than the right one > 0 : the left string is larger than the right one *Example:* `char name1[20] = "John";` `char name2[20] = "John";` `char name3[20] = "Zack";` `// To compare whether exactly the same` `if (strcmp(name1,name2)== 0) { …. }` `cout<<strcmp(name1,name2); // Out: 0` `cout<<strcmp(name1,name3); // Out: -1` `cout<<strcmp(name3,name1); // Out: 1`	**C++ string has similar method working the same** *Example:* `string name1 = "John";` `string name2 = "John";` `string name3 = "Zack";` `// To compare whether exactly the same` `if (name1.compare(name2)== 0){….}` `if (name2.compare(name1)== 0){….}` `cout<<name1.compare(name2); // 0` `cout<<name1.compare(name3); // -1` `cout<<name3.compare(name1); // 1`

C vs C++ Strings (5)

	C String	C++ String
Concatenation	`char name[20] = "John";` `char greeting[20] = "Hello ";` *// To merge or concatenate the name to greeting* **`strcat`**`(greeting,name);` *// Result, greeting now holds "Hello John"* *// Cannot use the + operator* `greeting = greeting + name;`	`string name = "John";` `string greeting = "Hello ";` *// Using method* **`greeting.append(name);`** *// Using the + or += operator* **`greeting = greeting + name;`** **`greeting += name;`**

C vs C++ Strings (6)

	C String	C++ String
String to number conversion	`char strNumber[5] = "20";` `char strFloat[5] = "1.5";` `int num1;` `double num2,result;` *// Cannot perform arithmetic on string* `result = strFloat - 1;` *// Convert a string to an integer and a double* **`num1= atoi(strNumber);`** **`num2= atof(strFloat);`**	`string strNumber = "20";` `string strFloat = "1.5";` `int num1;` `double num2;` *// Convert a string to an integer and a double* **`num1= stoi(strNumber);`** **`num2= stod(strFloat);`**
Number to string conversion	`int number = 210;` `char strNum[5];` `int numDigit;` *// Example: Determine number of digits* *// Convert a number to string in base-10* **`itoa(number,strNum,10);`** *// Determine the string length* **`numDigit= strlen(strNum);`**	`int number = 210;` `string strNum;` `int numDigit;` *// Convert a number to string* **`strNum = to_string(number);`** *// Determine the string length* **`numDigit = strNum.length();`**

C vs C++ Strings (7)

	C String	C++ String
C string to C++ string conversion (and vice versa)	• **Why? You store your data as C string, but you want to manipulate the data as C++ string.** • **Done by creating a C++ string object** ***Example:*** *Convert to C++ string in order to use the + operator for concatenation* `char name[20] = "John";` `char greeting[20] = "Hello ";` **`cout<<( string(greeting)`** **`+ string(name) )<< endl;`** *// Output: Hello John*	• **Done by using the method c_str()** ***Example:*** *Convert string to integer using the C-string function. However, the data stored as C++ string* `string strNum = "25";` `int value;` `value = atoi( strNum.c_str() );` `cout<< (value * 2 )<< endl;`