Part I — C++

Based on lectures by Brian

Notes taken by Dexter Chua

Sem 2 2017-2018

These notes are not endorsed by the lecturers, and I have modified them (often significantly) after lectures. They are nowhere near accurate representations of what was actually lectured, and in particular, all errors are almost surely mine.

Contents

1 Print out the code

1.1 Hello world!

```
#include<stdio.h>
#include<iostream>
// A comment
int main(void)
{
printf("Hello␣World\n");
return 0;
}
```

2 Template

```
#include <iostream>
using namespace std;
int main(){
          float hkd;
          float result;
          cin >>hk;
          result=hkd*14.2
          cout <<result; // print out the result
}
```

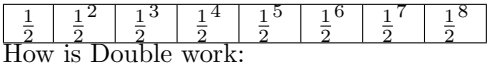
3 Data type

3.1 Variable and Constant

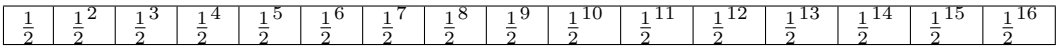
3.1.1 Numerical

- int: Integer type
- flaot, double:

Remark. How is Float works:



How is Double work:



Example.

3.1.2 Character

- char:

3.1.3 Logic

- bool:Boolean (true, false)

3.1.4 Other

- void:

4 Basic Operators

4.1 Type of Operators

4.1.1 Number Operator

4.1.2 Comparative Operator

This operators structure will return to boolean

```
\\Equality operators
== // equal to
!= // not equal to

\\Relational operators
>  // greater than
>=  // greater and equal than
<  //
<=  //
```

4.1.3 Logical Operator

This operators structure will return to boolean

```
!  // not
&&  // and
||  // or
```

Example. Given integer variable i,j and k, what are the outputs when running the program fragment below?

```
k = (i=2) \&\& (j=2) ;
cout << i << j << endl; /* 2 2 */

k = (i=0) \&\& (j=3) ;
cout << i << j << endl; /* 0 2 */

k = i ||  (j=4) ;
cout << i << j << endl; /* 0 4 */

k = (i=2) || (j=5) ;
cout << i << j << endl; /* 2 4 */
```

Answers:

4.1.4 Conditional Operator

4.1.5 Comma operator

5 Method

6 Flow control

6.1 If else statement

if must in first part else if else must in last part

```
if (logical_expression){

    statement;
    statement;

}
else if (logical_expression){

     statement;
     statement;

}

else{

   statement;
   statement;

}
```

6.2 Switch statement

What is Switch statement look like:

```
switch (expression) {
        case constant-expr1: statement1
        case constant-expr2: statement2
        ...
        ...
        case constant-exprN: statmentN
        default: statement
}
```

Example. Here is the example for using switch statement:

```
while ((c = getchar()) != EOF) {  /* get a char */
      switch (c) {
        case ?0?: case ?1?: case ?2?: case ?3?: case ?4?:
        case ?5?: case ?6?: case ?7?: case ?8?: case ?9?:
         digit_count++;   /* no braces is needed */
         break;
        case ? ?: case ?\n?: case ?\t?:
         white_character_count++;
         break;
        default:
```

```
            other_character_count++;
            break;
    }
}
```

7 Loop

7.1 While Loop

7.2 Do-while Loop

7.3 For Loop

8 Functions

8.1 Introduction

Example.

```
 #include <iostream> using namespace std;
void printHello(int n){
  for (int i=0;i<n;i++)
      cout <<"Hello" <<endl;
}
void main() {
  printHello(10);
}
```

8.2 Calling functions

Example.

```
 #include <iostream> using namespace std;
void printHello(int n){
  for (int i=0;i<n;i++)
      cout <<"Hello" <<endl;
}

 void main() {
int x=1;
  printHello(x);
  printHello(x+3);
  printHello(10);
}
```

8.3 Multiple parameters

Example.

```
#include <iostream>
using namespace std;
int maxValue(int a, int b){
      int m=a;
      if (b>a)
               m=b;
      return m;
}

void main() {
       int x , y=4, z=1;
       x = maxValue(4,2);
}
```

9 Global and local variable

9.1 Introduction

9.2 Example

Example.

```
#include <iostream>
using namespace std;
int num1=4;
int num2=9;
int maxValue(int a, int b){ /*  Local(maxValue)  */
        int m=a;
        if (b>a)
        m=b;  /*  Local(maxValue)  */
        return m;
}

void main() {
    int x;  /* Local(main) */
    x = maxValue(num1,num2); /* Global    */
}

Global
```

9.3 Parameters Passing: Pass-by-value

10 Class

10.1 Constructor

10.2 this