UCLA Samueli Computer Science

CS31 Week 3 Discussion

Fall 2021, Section 1C

Mingyu Derek Ma mdma@ucla.edu

Thanks Muhao Chen, Rosa Garza for their shared content

https://derek.ma/cs31 for slides and other discussion materials

Reminder

- Project 3 warmup: Oct 20 Wed 11pm

Loop

- Why do we use a loop?
  - When is task is needed to be repeated
  - We don’t need to repeat code
- What should you take care of in the loop?
  - Initialization
  - Stay-in-loop condition / Termination condition
  - Iteration progress / increment step

while loop

```
while (condition) {
	statements;
}
```

- Repeat
  - Check stay-in-loop condition
    - If condition is not met, stop and run following code after the loop
  - Perform loop body code block

do while loop

```
do {
        statements;
} while (condition);
```

- Run loop body
- Repeat
  - Check stay-in-loop condition
    - If not met, stop
  - Run loop body

- Will always execute code block inside at least once
- Need the semicolon at the end

while VS do while

- While: check condition before executing the body
- Do While: check condition after executing the body

```
int main(){
  int n=3, i=4, result=1;
  while (i <= n) {
    result *= i; i++;
  }
  cout<<result;
}
```

```
int main(){
  int n=3, i=4, result=1;
  do {
    result *= i; i++;
  } while (i <= n);
  cout<<result;
}
```

while VS do while

- While: check condition before executing the body
- Do While: check condition after executing the body

```
int main(){
  int n=3, i=4, result=1;
  while (i <= n) {
    result *= i; i++;
  }
  cout<<result;
}
```

- The termination condition is False, not even entered the loop
- Output: 1

```
int main(){
  int n=3, i=4, result=1;
  do {
    result *= i; i++;
  } while (i <= n);
  cout<<result;
}
```

- Execute loop body first, then determine the termination condition is False
- Output: 4

Avoid infinite loop

- Make sure we are making progress towards the termination condition
- Check whether the variable related to the termination expression is updated

```
int main(){
  int n=3, i=4, result=1;
  while (i <= n) {
    result *= i;
    //i++;
  }
  cout<<result;
}
```

for loop

```
for (init; condition; increment)
{
        statements;
}
```

Program example 1

- Init allows you to declare and initialize any loop control variables
- Condition is a boolean expression, allows you to decide when to terminate the loop

- Initialize step executed first
- Repeat
  - Check stay-in-loop condition
    - If not met, stop
  - Run loop body
  - Perform increment

for loop rewrite to while loop

```
for (init; condition; increment)
{
        statements;
}
```

```
for (int i=2; i<=n; ++i) {
        return *= 1;
}
```

The above ones are the same as the below ones

```
init
while (condition) {
        statements;
        increment
}
```

```
int i=2
while (i<=n) {
        result *= i;
        ++i;
}
```

for loop rewrite to do...while loop

```
for (init; condition; increment)
{
        statements;
}
```

- Can we find a equivalent do-while loop for a for loop?
- These three programs have the same logic

```
init
do {
        if (!condition)
                break;
        statements;
        increment;
} while (True);
```

```
init
if (condition) {
        do {
                statements;
                increment;
        } while (condition);
}
```

char

- Character type char is encoded using an integer representation of 1 byte
  - 1 byte = 8 bits
  - 2^8 = 256 character can be represented
- ASCII
  - Range: 0 to 255 inclusively
  - ASCII is the encoding schema
  - ‘ ’ is encoded as 32
  - ‘A’ is encoded as 65
  - ‘a’ is encoded as 97
  - ‘+’ is encoded as 43
  - ‘Z’ is encoded as 90
  - ‘z’ is encoded as 122

Arithmetic and relational operations on char

- 'a' < 'b' is true
- '4' > '3' is true
- '6' <= '2' is false
- ‘F’ – 5 is ‘A’
- Lower case letters is greater than its upper cases
- ‘x' + (‘A’ – ‘a’) is ‘X’
- ‘Y’ – (‘Z’ – ‘z’) is ‘y’
- ‘a’ – 32 is ‘A’

#include <cctype>

- #include <cctype> provides several useful functions for char
- isdigit(char c): Is c a digit?
- islower(char c): Is c lower case?
- isupper(char c): Is c upper case?
- isalpha(char c): Is c alphabetic?
- tolower(char c): Convert c to lower case
- toupper(char c): Convert c to upper case

char

- Use single quote to represent a single character
- Use double quote to represent a string
- Special characters are delineated by a backslash \
  - They are two-character sequences for escape codes
  - Common escape codes
    - \t denotes tab
    - \\ denotes a backslash
    - \” denotes a double quote
    - \’ denotes a single quote
    - \n denotes end-of-line
    - \0 end of string (NULL)

Program example 2

String

Program example 3

- String is a class in C++
  - We will introduce class in later lectures
  - It is similar to a data type, but more powerful
  - It can have its own functions and attributes
  - A string stores a sequence of characters
  - A string terminated by a null (\0) character
- To use string, we need to #include <string>
- size()
  - Get the number of characters (\0 does not count)
- Access a certain character in string
  - s[i] or s.at(i) to access the (i+1)th character
  - Index of character in string starts from 0
  - Caution for undefined behavior (beofre C++ 11): i in s[i] has to be =< s.size() - 1

```
string s = “ab cd”
s.size() // 5
s.at(1) // ‘b’, it’s char type
s[1] // ‘b’, also char type
```

Thank You