Xi'an Jiaotong-Liverpool University
西交利物浦大学

CPT104 - Operating Systems Concepts

Lab 2

Branching

If, Else, True and False

- In C, the following are FALSE: `0 0.0 '\0'`
  - Other than that, they are TRUE

unlike Java, there is no boolean type, instead use special value in int, double, char for false: zero, zero.zero and **null** character

```
#include <stdio.h>

int main() {
    int goodWeather;
    goodWeather = 1;
    if(goodWeather) {
        printf("The weather is good\n");
    } else {
        printf("The weather is bad\n");
    }
    return 0;
}
```

go to Codecast and run the code
if / else works just like in Java

change this to whatever type / value you like, what gets printed ?

Comparison Operators

- Comparison operator: == != < > <= >=

similar to Java

```
#include <stdio.h>

int main() {
    int a = 5;
    int b = 2;
    int result;
    result = a == b;
    if(result) {
        printf("That is TRUE\n");
    } else {
        printf("That is FALSE\n");
    }
    return 0;
}
```

change this to whatever type / value you like, what gets printed ?

change this to whatever comparison operator you like,
what gets printed ?

Comparing Floating Point Numbers

- Careful with precision issues, decimals stored as binary have limitation

```
#include <stdio.h>

int main() {
    double a = 5.00000000000000001;
    double b = 5.00000000000000000;
    int result;
    result = a == b;
    if(result) {
        printf("That is TRUE\n");
    } else {
        printf("That is FALSE\n");
    }
    return 0;
}
```

go to Codecast and
run the code
surprising result?!

try reducing the zero
on both of them,
at what point the
printing result
becomes different?

Logical AND

logical operators behave similar to Java

- 1 && 1 is 1, 1 && 0 is 0, 0 && 1 is 0, 0 && 0 is 0

```c
#include <stdio.h>

int main() {
    int sunny = 1;
    int vacation = 1;
    int sunnyAndVacation= sunny && vacation;
    if(sunnyAndVacation) {
        printf("Aw yeah!\n");
    } else {
        printf("Oh noes...\n");
    }
    return 0;
}
```

Logical OR

- 1 || 1 is 1, 1 || 0 is 1, 0 || 1 is 1, 0 || 0 is 0

```
#include <stdio.h>

int main() {
    int sunny = 1;
    int vacation = 0;
    int sunnyOrVacation = sunny || vacation;
    if(sunnyOrVacation) {
        printf("Ah well.\n");
    } else {
        printf("Oh noes...\n");
    }
    return 0;
}
```

Logical NOT

- !1 is 0, !0 is 1

```
#include <stdio.h>

int main() {
    int sunny = 0;
    int vacation = 1;
    int notSunnyAndVacation = !sunny && vacation;
    if(notSunnyAndVacation) {
        printf("It's cloudy but at least holiday\n");
    }
    return 0;
}
```

Branch using complex logical conditions

suppose you can work when you're not younger than 18 and less than 65 of age

- Combine logical conditions

```c
#include <stdio.h>

int main() {
    int age;
    printf("What is your age?\n");
    scanf("%d", &age);
    if(age >= 18 && !(age >= 65)) {
        printf("You are in labor force.\n");
    } else {
        printf("You are not in labor force\n");
    }
    return 0;
}
```

run in Codecast and input different numbers

Looping while Checking Elements

- Write a program that reads a number n, ask the user to enter n numbers, and then print the **sum** of those n numbers ***which are even***

```c
#include <stdio.h>
int main() {
    int i, n, num;
    int sum = 0;
    scanf("%d", &n);
    for(i=0; i<n; i++) {
        scanf("%d", &num);
        if(num % 2 == 0) {
            sum += num;
        }
    }
    printf("%d\n", sum);
    return 0;
}
```

checking if user input is even

then sum it up

(else do nothing)

Thank you for your attention !

- In this lab, you have learned:
  - TRUE and FALSE in C
  - Branching using if, else
  - Logical operator AND, OR, NOT, and their combinations
  - Looping while checking elements

- Reference book: chapter 2, section 2.6, 2.11, chapter 3