

# COMP1917: 03 Making Decisions

Sim Mautner

*simm@cse.unsw.edu.au*

August 1, 2016

# Announcements

- E-mail (try: [s.mautner@unsw.edu.au](mailto:s.mautner@unsw.edu.au))
- Consultations

# References

- Moffat, Chapter 3: Making Decisions

# Making Decisions

- Different behaviour in different situations.

## Driving Example v1

Write an application which asks the user to enter their age. If they are at least 16 years old, display “You can drive”. Then, whether or not they can drive, display “Have a nice day.”

## Driving Example v1

Write an application which asks the user to enter their age. **If** they are at least 16 years old, display “You can drive”. Then, whether or not they can drive, display “Have a nice day.”

# Driving Example v1: Step by Step

- 1 Print "How old are you?"
- 2 Read in their age.
- 3 If they are  $\geq 16$ :
  - 1 Print "You can drive."
- 4 Print "Have a nice day."

# Driving Example v1

```
#include <stdio.h>
```

```
int main(int argc, char *argv[]) {  
  
    printf("How old are you? ");  
    int age = 0;  
    if(age >= 16) {  
        printf("You can drive.\n");  
    }  
    printf("Have a nice day.\n");  
}
```



## Detour: Definition of Constant Values

- Using the same value numerous times in a program becomes high maintenance if the value changes and needs to be changed in many places. (You may miss one!!)
- Other developers may not know (or you may forget) what that number means.
- In this case, the minimum driving age is 16.

```
#define MIN_DRIVING_AGE 16
```

- **Note:** No semi-colon at the end of this line.

## Detour: Definition of Constant Values

```
#include <stdio.h>
```

```
#define MIN_DRIVING_AGE 16
```

```
int main(int argc, char *argv[]) {  
  
    printf("How old are you? ");  
    int age = 0;  
    if(age >= MIN_DRIVING_AGE) {  
        printf("You can drive.\n");  
    }  
    printf("Have a nice day.\n");  
}
```

## Driving Example v2

Write an application which asks the user to enter their age. If they are at least 16 years old, display “You can drive”. Otherwise, display “You cannot drive.”. Then, whether or not they can drive, display “Have a nice day.”

## Driving Example v2

Write an application which asks the user to enter their age. **If** they are at least 16 years old, display “You can drive”. **Otherwise**, display “You cannot drive.”. Then, whether or not they can drive, display “Have a nice day.”

## Driving Example v2: Step by Step

- 1 Print "How old are you?"
- 2 Read in their age.
- 3 If they are  $\geq 16$ :
  - 1 Print "You can drive."
- 4 Otherwise:
  - 1 Print "You cannot drive."
- 5 Print "Have a nice day."

## Driving Example v2

```
#include <stdio.h>
```

```
#define MIN_DRIVING_AGE 16
```

```
int main(int argc, char *argv[]) {  
  
    printf("How old are you? ");  
    int age = 0;  
    if(age >= MIN_DRIVING_AGE) {  
        printf("You can drive.\n");  
    } else {  
        printf("You cannot drive.\n");  
    }  
    printf("Have a nice day.\n");  
}
```

# More Conditions and Input Validation

- Sometimes we want to consider more than 2 options for paths.
- In the case of the driving scenario, we want to make sure that the age is  $\geq 0$  and  $\leq 120$ .

## Driving Example v3

```
printf("How old are you? ");
int age = 0;
if(age < 0) {
    printf("Invalid input.\n");
} else if(age < MIN_DRIVING_AGE) {
    printf("You cannot drive.\n");
} else if(age <= MAX_DRIVING_AGE) {
    printf("You can drive.\n");
} else {
    printf("Invalid input.\n");
}
printf("Have a nice day.\n");
```



# Conditions

<b>C Symbol</b>	<b>Math Symbol</b>	<b>Explanation</b>
<	<	Less than
<=	≤	Less than or equal to
>	>	Greater than
>=	≥	Greater than or equal to
==	=	Equal to
!=	≠	Not equal to

# Nested If Statements

```
if(age >= MIN_DRIVING_AGE) {  
    if(age <= MAX_DRIVING_AGE) {  
        printf("You can drive.\n");  
    }  
}
```

# Logical Operators

- Used when we want to check multiple conditions in a single if statement.

<b>Symbol</b>	<b>Meaning</b>	<b>Explanation</b>
&&	AND	Both statements must be true.
	OR	One or more of the statements must be true.
!	NOT	The statement must be false.

## Nested If Statements

```
if(age >= MIN_DRIVING_AGE) {  
    if(age <= MAX_DRIVING_AGE) {  
        printf("You can drive.\n");  
    }  
}
```

```
if(age >= MIN_DRIVING_AGE && age <= MAX_DRIVING_AGE) {  
    printf("You can drive.\n");  
}
```

# Making Decisions Summary

```
if(/* condition 1 */) {  
    // Do stuff  
} else if(/* condition 2 */) {  
    // Do something else  
} else if(/* condition 3 */) {  
    // Do something completely different  
} else {  
    // In all other cases do this.  
}
```

Symbol	Meaning	Explanation
&&	AND	Both statements must be true.
	OR	One or more of the statements must be true.
!	NOT	The statement must be false.

# Making Decisions Summary

<b>C Symbol</b>	<b>Math Symbol</b>	<b>Explanation</b>
<	<	Less than
<=	≤	Less than or equal to
>	>	Greater than
>=	≥	Greater than or equal to
==	=	Equal to
!=	≠	Not equal to

## Try It Yourself

- At a certain theme park has rides which anyone over 120cm can go on (“Yellow rides”), rides which anyone over 130cm can go on (“Green rides”) and rides which anyone over 140cm can go on (“Red rides”). Write an application which asks the user to enter their height to the nearest cm and displays which kinds of rides they can go on.
- Write an application which asks the user for two numbers and a letter.
  - ▶ If the letter is ‘a’, add the two numbers and print the result.
  - ▶ If the letter is ‘s’, subtract the second number from the first and print the result.
  - ▶ If the letter is ‘m’, multiply the two numbers and print the result.
  - ▶ If the letter is ‘d’, divide the first number by the second number and print the result.