Intro. Computing with the C Programming Language

# Conditionals and Recursion

Shin Hong

14 September 2023

# Programming Conditional Behaviors

Write a program that prints “Odd” if a variable has an odd number or, prints “Even”

# Conditional Statement

- A conditional statement determines the execution of other statements by the evaluation of an expression
  - if statement,  if-else statement
  - switch statement
  - while statement, for statement, do-while statement

- A condition expression typically checks equality, or inequality of expressions

- Ex. if-statement

```
if ( expr ) {
   stmt1 ;
   stmt2 ;
   ...
}
```

# Example

- `print_int.c`
  - if-statement
  - if-else statement
  - nested if-statement

# Comparison Operators

- A comparison operator is a binary operator connecting two operands of the same type
- A comparison operator produces either zero (false) or a non-zero value (true)
- example
  - `x == y`
  - `x != y`
  - `x > y`
  - `x < y`
  - `x >= y`
  - `y <= x`

# Return Statement

- The return statement terminates the function execution when it is executed

- For a function with a non-void type, the return statement determines the return value

- Ex. `triangle.c`
  - input validity checking
  - return value

# Receiving User Input

- `scanf` can be used to read an integer, double or char from user’s typing
  - example

```
int x ;
scanf(“%d”, &x) ;
```

# Recursion

- A function may call other functions and may call itself
- The process of a function calling itself is called recursion and such a function is called recursive
- Example
  - `countdown.c`
- A recursive function consists of the recursive case and the base case
  - base case returns without a recursive call
  - recursive case calls the function itself with a different argument
  - an infinite recursion happens if the program has an error that the base case is not properly executed

# Stack

- Every time a function gets called, it creates a new instance that contains the function’s local variables and parameters
  - a list of such instances are called stack
  - each instance is called stack frame

- Ex. `sum_of_sums.c`

Intro. Computing with the C Programming Language

# More on Functions

Shin Hong

18 September 2023

# Return value of function

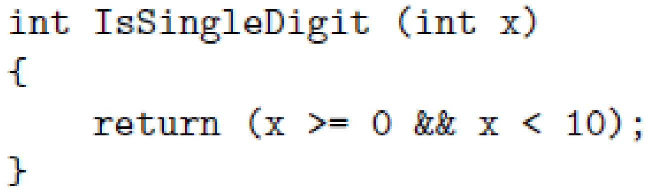
- a function whose return type is `void` does not return any value to the caller
  - the return statement in such a function just finish the function execution

- A non-void function must execute a return statement in any execution path
  - the compiler throws an error if there seems any possibility that a function does finish with a return statement
  - ex. `absolute_value.c`

# Boolean variables

- The result of a comparison operator is a short value
  - 0 for False, and a non-zero value for True
  - C does not support a primitive type of Boolean values

- Logical operators compose compound expressions
  - AND. e.g., `x > 0 && y > 0`
  - OR.  e.g., `x > 0 && y > 0`
  - NOT. e.g., `!(x > 0)`
  - Ex.



```
int IsSingleDigit (int x)
{
    return (x >= 0 && x < 10);
}
```

# Returning from `main()`

- The main function is invoked by OS as a starting point of a program execution
- The return value of the main function defines an error code of a program which notifies to OS whether the program successfully terminates, or it terminates due to an error
  - 0 means a successful termination
  - a non-zero value means an error
- Ex. `exit_code.c`