Intro. Computing with the C Programming Language

Looping and Branching

- PIC: Ch. 4. Program Looping
- PIC: Ch. 5. Making Decision
- GCM: Ch. 4. Statements

Shin Hong

9 November 2023

For-loop Statement

- Basic structure

```
for (init_expr; loop_cond ; loop_expr)
    statements
```

- Variation examples

```
for (i=0, j=100 ; i < 10 ; ++i, j=j-10)
   ...
```

```
for ( ; j != 100 ; ++j)
   ...
```

```
for (int n = 1, tri = 0; n <= 200; ++n)
   tri += n ;
```

Other Loop Statements

- While-loop statement

```
while (loop_condition) {
   statements
}
```

- Do-while-loop statement

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

- Ex. `loop.c`

Looping Control

- `Break` statement
  - leave the loop when executed
- `Continue` statement
  - bypass the statements after the continue statement, and continue the loop
- Ex. `fivenum.c`

If Statement

- Basic structure

```
if (expression)
   statements1
```

- Variations

```
if (expression)
   statements1
else
   statements2
```

```
if (expression1)
   statements1
else if (expression2)
   statements2
else
   statements3
```

Switch-case Statement

- Basic structure

```
switch ( expression ) {
     case value1:
               program statement
               program statement
                    ...
               break;
     case value2:
               program statement
               program statement
                    ...
               break;
          ...
     case value_n:
               program statement
               program statement
                    ...
               break;
     default:
               program statement
               program statement
                    ...
               break;
}
```

- Ex. `calc.c`

Compound Relational Test

- Logical “And” operator
  - `A && B` is true if and only if `A` is true and `B` is true
- Logical “Or” operator
  - `A || B` is true if and only if `A` is true or `B` is true
- Logical “Negation” operator
  - `!A` is true if and only if `A` is false
- Ex. `leapyear.c`

Conditional Operator

- Basic structure

```
condition ? expression1 : expression 2
```

- Examples

```
abs_value = (x >= 0) ? x : x * -1 ;

max = (a > b) ? a : b ;

sign = (a > 0) ? 1 : ((a == 0) ? 0 : -1) ;
```