Problem Solving & Program Design in C

Chapter 4:
Selection Structures: If and Switch Statements

Control Structures

- Compound statement or block of code
- Use {…} to contain compound statement
- Variables declared in {…} are not available outside {Vegas}
- Declare variables before { and change in {…} to use after }
- Selection control structures chose which code to execute based on conditions in the program

Conditions

- Simple conditional expressions
- Contain variables, constants and relational operators
- Relational expressions are 1 if true and 0 if false
- English terms

TABLE 4.2 Sample Conditions

Operator	Condition	English Meaning	Value
<=	`x <= 0`	`x` less than or equal to 0	1 (true)
<	`power < MAX_POW`	`power` less than `MAX_POW`	0 (false)
>=	`x >= y`	`x` greater than or equal to `y`	0 (false)
>	`item > MIN_ITEM`	`item` greater than `MIN_ITEM`	1 (true)
==	`mom_or_dad == 'M'`	`mom_or_dad` equal to `'M'`	1 (true)
!=	`num != SENTINEL`	`num` not equal to `SENTINEL`	0 (false)

Logical Operators

- Logical expressions
- Used for compound relational expressions
- AND &&, OR || and NOT !
  - x < y && y > z
  - x < y || y > z
  - !(x < y)
  - !(any non-zero value or expression) = 0
  - !(0) = 1
- Exclusive OR

Operator Precedence

- Arithmetic follows algebraic rules
- Add relational and Logical

TABLE 4.6 Operator Precedence

Operator	Precedence
function calls	highest
! + - & (unary operators)	
* / %	
+ -	
< <= >= >	
== !=	
&&	
||	
=	lowest

Short-Circuit Evaluation

- Feature that increases performance and provides the same result
- For AND: x < y && y > z
  - If first relational expression , x < y, is false, then the compound expression is false
- For OR: x < y || y > z
  - If first relational expression, x < y, is true, then the compound expression is true

Comparing Characters

- Relational and logical operators work with chars too.
- Chars are binary numbers to a computer
- Alphabet chars are in order in ASCII table
  - ‘ ‘ = 32
  - ‘0’ = 48 and ‘9’ = 57
  - ‘A’ = 65 and ‘Z’ = 90
  - ‘a’ = 97 and ‘z’ = 122

```
int isLower(char c){
   return (c >= 'a' && c <= 'z');
}
```

Comparing Floating-Point Numbers

- Floats and double can have minor precision errors due to arithmetic calculation
- Must allow for some error
- Consider currency calculations

```
int main(){

   double x = 10.0;
   double y = 10.000003;
   double e = 0.009;

   if(x == y)
      printf("They are equal");

   if(fabs(x-y) < e)
      printf("They are approximately equal");

   return 0;
}
```

Relational and Logical Assignment

- Can use int to store the result of a relational or logical expression

```
int a = (x < y);
int b = (x < y && y > z);
int c = (x < y || y > z);
int in_range = (n > -10 && n < 10);
int is_letter = (c >= 'A' && c <= 'Z') ||
                (c >= 'a' && c <= 'z')
```

Complementing a Condition

- DeMorgan's Law – reverse all symbols in expression
  - !(<) = (>=) and !(>) = (<=)
  - !(==) = (!=) and !(!=) = (==)
  - !(&&) = (||) and !(||) = (&&)
- Examples
  - !(x < y) = (x >= y)
  - !(x == y) = (x != y)
  - !(x < y && y > z) = (x >= y || y <= z)
  - !(x < y || y > z) = (x >= y && y <= z)

The If Statement

- Basic if statement decides between two alternatives

```
if(x < y)
   printf("x is less than y\n");        // Executed if true
else
   printf("x is NOT less than y\n"); // Executed if false
```

- BaBasic if statement with one alternative

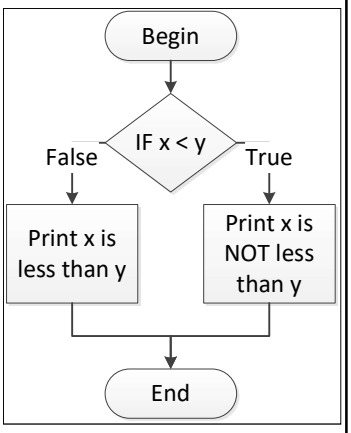
```
if(x < y)
   printf("x is less than y\n");        // Executed if true
```

Style and design

- Flow charts
- Pseudocode
- Tabbing statements

```
If x is less than y
   Print x is less than y
Else
   Print x is NOT less than y
```

```
if(x < y)
   printf("x is less than y\n");        // Executed if true
else
   printf("x is NOT less than y\n"); // Executed if false
```



If Statements with Compound Statements

- If statements can execute more than one code statement per condition

```
if(x < y){
   // Block executed if true
   printf("x is less than y\n");
   printf("y is greater than or equal to x\n");
}
else{
   // Block executed if false
   printf("x is NOT less than y\n");
   printf("y is less than or equal to  x\n");
}
```

Nested If Statements

- If statements can contain other If statements
- Else is paired with the nearest unpaired If
  - Can be overridden with {}
  - Can also have logical expressions
  - Can also use {…} for compound code statements

```
if(x < y)
   if(y > z)
      printf("x is less than y AND y is greater than z\n");
   else
      printf("x is less than y AND y is NOT greater than z\n");
else
   if(y > z)
      printf("x is NOT less than y but y is greater than z\n");
   else
      printf("x is NOT less than y AND y is NOT greater than z\n");
```

Multiple-Alternative If Statements

- If statements can have more than two alternatives using else if
- Can also use {…} for compound code statements
- Can also nest in else if chain and use {} to control execution

```
if(x < 5)
   printf("x is less than 5\n");
else if(x < 10)
   printf("x is less than 10\n");
else if(x < 15)
   printf("x is less than 15\n");
else
   printf("x is NOT less than 15\n");
```

The Switch Statement

- Similar to If-Else
- Switch on condition in data
  - Can use char, int or enum
- Case for each alternative
- Break stops execution of multiple cases

```
enum Colors {Red, Green, Blue};
…
Colors color = Green;
switch(color){
   case Red:
      printf("Color is Red\n");
      break;
   case Green:
      printf("Color is Green\n");
      break;
   case Blue:
      printf("Color is Blue\n");
      break;
   default:
      printf("Unknown color\n");
}
```

Common Programming Errors

- 5 < x < 10 won't work
  - x > 5 && X < 10 will
- if(x = 5) won't work – always true
  - if(x == 5) works
- Logical ordering

```
if(x > 0)
   …
else if(x > 5)
   …
else if(x > 10)
   …
else
   …
```