ESc 101: Fundamentals of Computing

Lecture 6

Jan 13, 2010

Range of Values for Different Types

`int`: -2^{31} to $+2^{31} - 1$

`char`: character \0 (null) to character ÿ

`float`: -2^{128} (approx) to $+2^{128}$ (approx)

`double`: -2^{1024} (approx) to $+2^{1024}$ (approx)

Recall: Basic Structure

```
main()
<statement-block>
```

<statement-block> has the form:

```
{
    <variable declarations>
    <statements>
}
```

Recall: Variable Declarations

`<variable declarations>` is a sequence of declarations:

```
<declaration-1>
<declaration-2>
      :
      :
<declaration-n>
```

Variable declaration: `<type> <name>;`

Variable Names

- <name> is a sequence of letters and digits starting with a letter.
- Example: `num1`, `num2new` (`3num` is invalid name)
- Capital letters are treated as different from small letters.
- Example: `num1`, `Num1`, `NUM1` are distinct variables names
- The symbol _ is treated as a letter.
- Example: `num_1`, `num_2_new`, `_num`
- Caution: Do not use names beginning with _ as compiler uses these names.

Statements

<statements> is a sequence of statements:

```
<statement-1>
<statement-2>
      :
      :
<statement-n>
```

A Single Statement

Each statement is one of the three kinds:

- Assignment statement
- Conditional statement
- Loop

Assignment Statement

Its form is:

```
<name> = <expression>;
```

- `<name>` is a variable name.
- `<expression>` is an expression whose result is stored in `<name>` by the statement.

Conditional Statement

Its form is:

```
if (<condition>)
   <statement-block-1>
else
   <statement-block-2>
```

- <condition> is an expression.
- If the value of <condition> is non-zero then <statement-block-1> is executed, otherwise <statement-block-2>.
- if a statement block has only a single statement, then the curly braces can be dropped.

Loop

Its form is:

```
for (<st-1>, ..., <st-n>; <condition>; <ste-1>, ..., <ste-m>)
   <statement-block>
```

- During the execution, first statements <st-1>, ..., <st-n> are executed.
- Then, if <condition> is non-zero, <statement-block> is executed.
- Then statements <ste-1>, ..., <ste-m> are executed.
- Then, if <condition> is non-zero, <statement-block> is executed, and so on.
- When the <condition> becomes zero, the execution goes past the loop.

Converting Sequence of Digits to Number - 1

```
main()
{
   int digit;
   int number;

   digit = (int) getchar() - 48;
   number = 0;

   for (; (digit >= 0) && (digit <= 9); ) {
      number = number * 10 + digit;
      digit = (int) getchar() - 48;
   }

   printf("%d\n", number);
}
```

Converting Sequence of Digits to Number - 2

```
main()
{
   int digit;
   int number;

   digit = (int) getchar() - 48;
   number = 0;

   for (; (digit >= 0) && (digit <= 9);
            digit = (int) getchar() - 48) {
      number = number * 10 + digit;
   }

   printf("%d\n", number);
}
```

Converting Sequence of Digits to Number - 3

```
main()
{
   int digit;
   int number;

   for ( digit = (int) getchar() - 48, number = 0;
         (digit >= 0) && (digit <= 9);
         digit = (int) getchar() - 48),
                number = number * 10 + digit );

   printf("%d\n", number);
}
```

Good Practices

One should include only simple statements inside the `for` loop brackets.

A better way of writing the same program:

```
main()
{
   int digit;
   int number;

   digit = (int) getchar() - 48;

   for (number = 0; (digit >= 0) && (digit <= 9); ) {

      number = number * 10 + digit;
      digit = (int) getchar() - 48;
   }

   printf("%d\n", number);
}
```