Structured Programming

- Primary data types and variables

Donglong Chen

With Thanks to Dr. Xin Feng and Dr. Haipeng Guo

Outline

- Values
- Primary data types
- Identifier
- Keywords
- Variables
- Declaration

Values

- There are different types of value
  - E.g.,
  - Age: 19
  - Gender: ’m’ or ’f’
  - Name: ” Tommy”
  - Weight: 82.5 (kg)
  - Time: 13:25:16

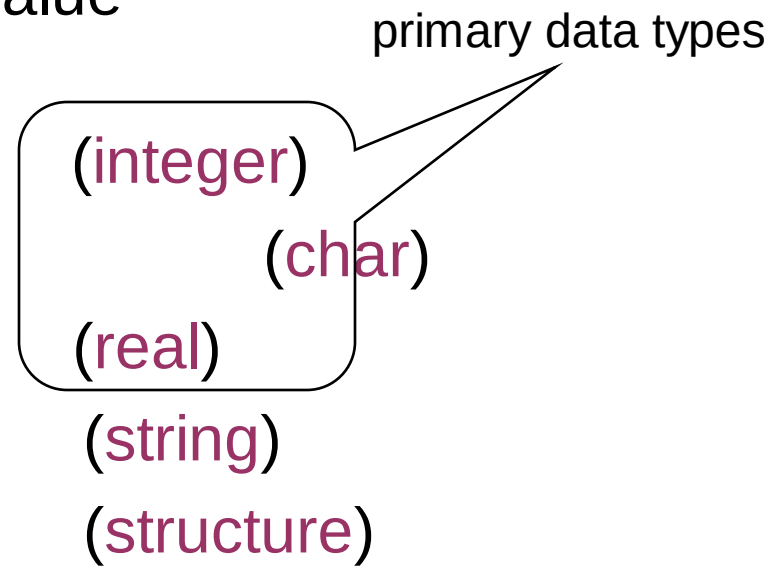
Recommend reading about how to pronounce punctuation in English:
https://simple.wikipedia.org/wiki/Punctuation
http://www.ruanyifeng.com/blog/2007/07/english_punctuation.html

Values

- There are different types of value
  - E.g.,
  - Age: 19 (integer)
  - Gender: ’ m’ or ’ f’ (char)
  - Weight: 82.5 (kg) (real)
  - Name: ” Tommy” (string)
  - Time: 13:25:16 (structure)

Values

- There are different types of value
  - E.g.,
  - Age: 19
  - Gender: ’ m’ or ’ f’
  - Weight: 82.5 (kg)
  - Name: “Tommy”
  - Time: 13:25:16



Primary Data Types

- int
  - Used to express the integer type. The biggest integer that can be expressed in a computer depends on the host computer (32 bits or 64 bits)
- char
  - Used to express the single characters. Each character corresponds to an integer between 0 and 127
- float
  - Real number (single precision float point)
- double
  - Real number (double precision float point)
- bool
  - A Boolean value which takes value 0 or 1

int

- int
  - A natural number (including 0), a negative number
  - E.g., 10, 20, 10000
  - Can be expressed in
    - Decimal (base-10)
    - Binary (base-2)
    - Hexadecimal (base-16)
    - Octal (base-8)

int

- 32 bits machine
  - 32 bits are used to express an integer value
- 64 bits machine
  - 64 bits are used to express an integer value

Question: Which one can express more integers?

int

- 32 bits machine
  - 32 bits are used to express an integer value
  - $-2^{31} \sim 2^{31} - 1$
- 64 bits machine
  - 64 bits are used to express an integer value
  - $-2^{63} \sim 2^{63} - 1$

int – Other int types

- short int
  - 2 bytes, $-2^{15} \sim 2^{15} - 1$
  - E.g., 12, 20
- long int
  - 4 bytes, $-2^{31} \sim 2^{31} - 1$
  - 20, 20L, 12, -2000, 0xffffL

int – Other int types

- unsigned int
  - 4 bytes, 0 ~ $2^{32} - 1$
  - E.g., 20, 12u, 0xffu
- long long int
  - 8 bytes, -2^{63} ~ $2^{63} - 1$
  - 20, 20LL,

Primary Data Types

- int
  - Used to express the integer type. The biggest integer that can be expressed in a computer depends on the host computer (32 bits or 64 bits)
- char
  - Used to express the single characters. Each character corresponds to an integer between 0 and 127
- float
  - Real number (single precision float point)
- double
  - Real number (double precision float point)
- _Bool
  - A Boolean value which takes value 0 or 1

char

- char
  - 1 byte
  - - 2^7 ~ 2^7 – 1
  - E.g., ’ a’, ’ 1’, ’ +’
- Attention
  - ’ 1’ is different from 1
  - ’ +’ is different from +
  - ’ a’ is different from a
  - ’ a’ is different from ” a”
- Every char corresponds to an integer code

Char - ASCII

- ASCII
  - American Standard Code for Information Interchange
  - Tables
    - https://www.ascii-code.com/

Char – Other char types

- unsigned char
  - 1 byte
  - $0 \sim 2^8 - 1$

Primary Data Types

- int
  - Used to express the integer type. The biggest integer that can be expressed in a computer depends on the host computer (32 bits or 64 bits)
- char
  - Used to express the single characters. Each character corresponds to an integer between 0 and 127
- float
  - Real number (single precision float point)
- double
  - Real number (double precision float point)
- _Bool
  - A Boolean value which takes value 0 or 1

float

- Float
  - 4 bytes
  - E.g., 1.2, 2.5e8 (Scientific notation. 2.5×10^{8})
  - $1.2 \times 10^{-38} \sim 3.4 \times 10^{38}$ (1.2e-38 ~ 3.4e38)
  - IEEE 754 standard: 1 bit sign, 8 bits exponent, 23 bits mantissa
- Double
  - 8 bytes
  - $2.2 \times 10^{-308} \sim 1.8 \times 10^{308}$ (2.2e-308 ~ 1.8e308)
  - IEEE 754 standard: 1 bit sign, 10 bits exponent, 53 bits mantissa

https://en.wikipedia.org/wiki/IEEE_754
https://en.wikipedia.org/wiki/Single-precision_floating-point_format

Primary Data Types

- int
  - Used to express the integer type. The biggest integer that can be expressed in a computer depends on the host computer (32 bits or 64 bits)
- char
  - Used to express the single characters. Each character corresponds to an integer between 0 and 127
- float
  - Real number (single precision float point)
- double
  - Real number (double precision float point)
- _Bool
  - A Boolean value which takes value 0 or 1

Identifiers (Variable Names)

- An identifier consists of a letter or underscore followed by any sequence of letters, digits or underscores
  - E.g., _Is, This_Is, A12, a23
- Names are case-sensitive! The following are unique identifiers:
  - Hello, hello,
  - whoami, whoAMI, WhoAmI
  - Are they same?

Identifiers (Variable Names)

- Names cannot have special characters in them
  - E.g., X=Y, J-20, #007 are invalid identifiers.
- C keywords (reserved words) cannot be used as identifiers.
- Choose identifiers that are meaningful and easy to remember.

Keywords

auto	break	case	char	const	continue	default	do
double	else	enum	extern	float	for	goto	if
int	long	register	return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned	void	volatile	while

Exercises

- Are these the valid variable names
  - _123
  - _abc
  - Example
  - Abc123
  - unsigned
  - int
  - a%b
  - 2example
  - Xx

Declaration and Assignment

- Every variable used in a program must declare its type
  - Format: TYPE variable_name _list;
  - E.g.,
    - int i;
    - float f;
    - double area;
    - unsigned int number;
    - int number, index, grade;

Declaration and Assignment

- The variables can be assigned values using the assignment operator
  - Format: variable_name = value;
  - E.g.,
    - i = 10;
    - f = 1.2;
    - area = 6.28 ;
    - area = f;

Declaration and Assignment

```
int i;
char c;
float f;
i = 28;
c = ' a' ;
f = 28.0;
```

declaration

assignment

Declaration and Assignment

```
int i1;
char 2c
float f;
i1 = 28.5;
2c =  ‘*’ ;
f = 28
```

What problems are there in the code?

Declaration and Assignment

```
int i;
char c;
float f;
i = 28;
c = 42;
f = 28;
```

Are these assignment OK?

Type Conversion

- C allows for conversions between the basic types, implicitly or explicitly.
- Explicit conversion uses the cast operator.

```
int x = 10;
float y = 3.14,z = 3.14;
y = (float)x;     /* y = 10.0 */
x = (int)z;       /* x = 3     */
x = (int)(-z);    /* x = -3 - rounded approaching zero */
y = x;           /* y = ??? */
```

cast operator

Implicit Conversion

- If the compiler expects one type at a position, but another type is provided, then implicit conversion occurs.

```
int x = 10;
float y = 3.14,z = 3.14;
y =(float)x;  /* y = 10.0 */
x =(int)z;    /* x = 3     */
x =(int)(-z); /* x = -3 - rounded approaching zero */
y = x;          /* y = -3.0 */
```

Implicit conversion

Implicit Conversion

- If the compiler expects one type at a position, but another type is provided, then implicit conversion occurs.

```
char c = 'a';
int i;
i = c;    /* i is assigned the ASCII code of 'a' */
```

Type: char

Type : int

Summary

- Basic data type
- Type casting