* High-level language
  - human friendly, allows programmer to express complex tasks with smaller code
  - values are given symbolic names
  - provides abstraction of underlying hardware (or ISA)

* General purpose

* Compiled, not interpreted
/* Our first sample C program to print some messages to the screen */

#include <stdio.h>

int main()
{
    int myValue = 6;
    printf("Ni hao! \n");
    printf("My lecture number=%d\n", myValue);

    return 0;
}

Basic Input/output (I/O):

* Output
  Display characters to the monitor (standard output)
Display characters to the monitor (standard output).

* Input
  Get characters from the keyboard (standard input)

1. Output (printf)
   - Requires formatted string that specifies
     1. Text to display
     2. Format to display the variable(s)
     3. Variables (or expression) to print

   e.g. printf("%d is even.", 22);

   1. int n = 100;
   2. printf("%d is odd", n+1);
printf("%d is odd", n+1);

Multiple variables:

int a = 2;
int b = 4;

printf("The numbers are %d and %d; their sum is %d\n", a, b, a+b);

The numbers are 2 and 4; their sum is 6 \n
Standard Input: (Scanf)

- Gets data from keyboard
- Performs type conversion from ASCII to specified format

E.g.: scanf("%c", &newchar);

Common formats:
e.g.: `scanf("%lf", &area);`
`scanf("%f", &radius);`
`sprintf("%f%f", &height, &weight);`

Common formats:

- `%d` 2's complement integer as decimal
- `%f` double printed as decimal
- `%u` unsigned integer printed as decimal