

```

/*
 * Converts distances from miles to kilometers.
 */
#include <stdio.h>          /* printf, scanf definitions */
#define KMS_PER_MILE 1.609 /* conversion constant */

int                          reserved word
main(void)
{
    double miles,           /* distance in miles */
    kms;                    /* equivalent distance in kilometers */

    /* Get the distance in miles. */
    printf("Enter the distance in miles> ");
    scanf("%lf", &miles);

    /* Convert the distance to kilometers. */
    kms = KMS_PER_MILE * miles;

    /* Display the distance in kilometers. */
    printf("That equals %f kilometers.\n", kms);

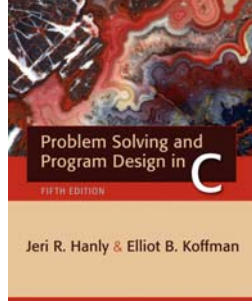
    return (0);
}

```

Diagram labels and arrows:

- preprocessor directive** points to `#include <stdio.h>` and `#define KMS_PER_MILE 1.609`.
- constant** points to `1.609`.
- reserved word** points to `int`, `main`, and `double`.
- variable** points to `miles` and `kms`.
- comment** points to `/* Converts distances from miles to kilometers. */` and `/* Get the distance in miles. */`.
- standard identifier** points to `printf` and `scanf`.
- special symbol** points to `*` in `KMS_PER_MILE * miles` and `*` in `printf("That equals %f kilometers.\n", kms);`.
- reserved word** points to `return`.
- punctuation** points to `(`, `)`, and `;` in `return (0);`.
- special symbol** points to `}`.

# C program

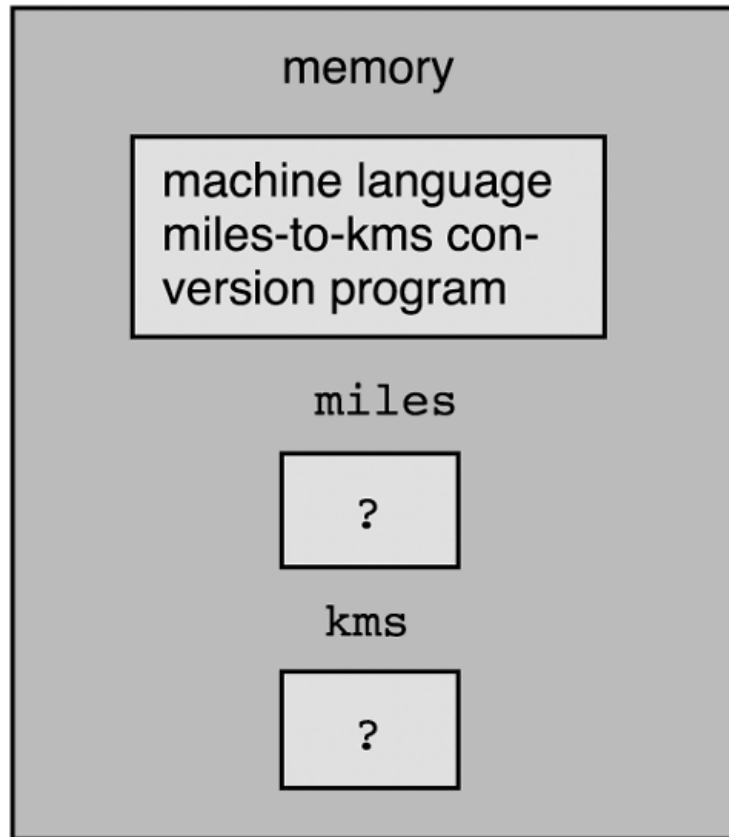


- The preprocessor part
- Definitions: Macro and Variable
  - Define the memory/data
  - Variable name and its data type
- Control Flow
  - Where should we start to execute? “main”
  - Reserved words
  - Function and blocks/scope of code

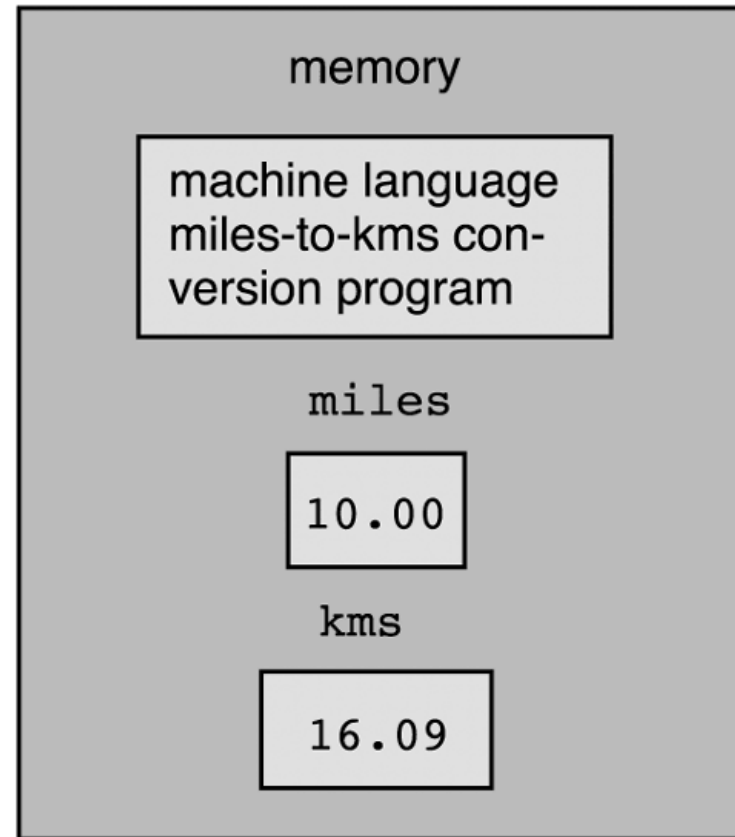
{ ... }

- { ... }
- { ... { ... { ... } ... } ... { ... } ... }
- int main (void) { ... }
- float myfunc (int x) { ...; return f; }
- { <declarations> <statements> }

# Memory Cell Concept Again



(a)



(b)