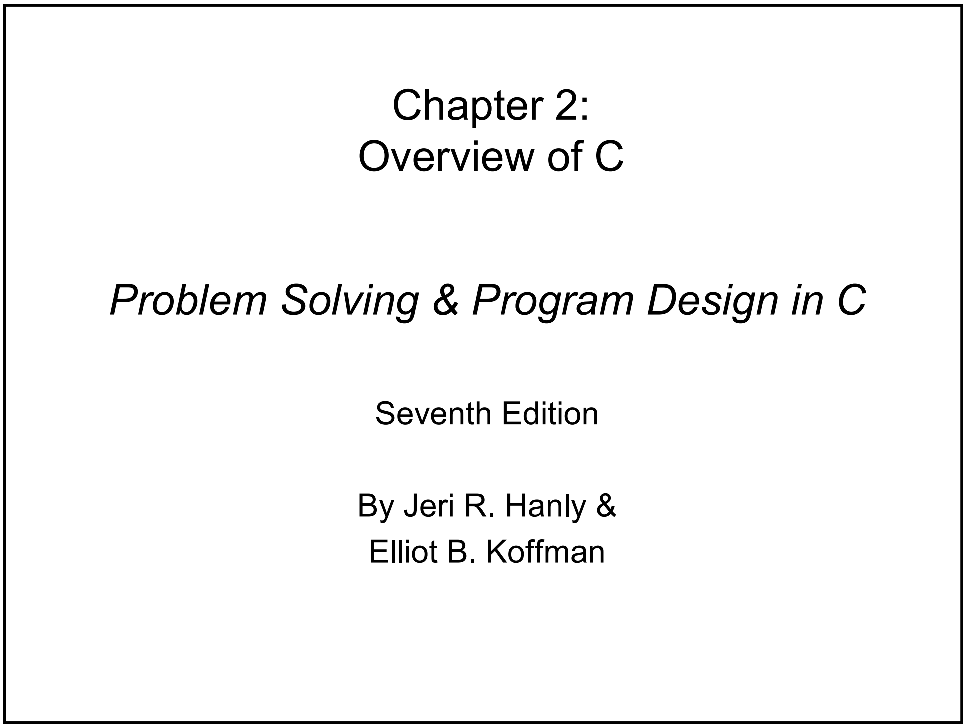


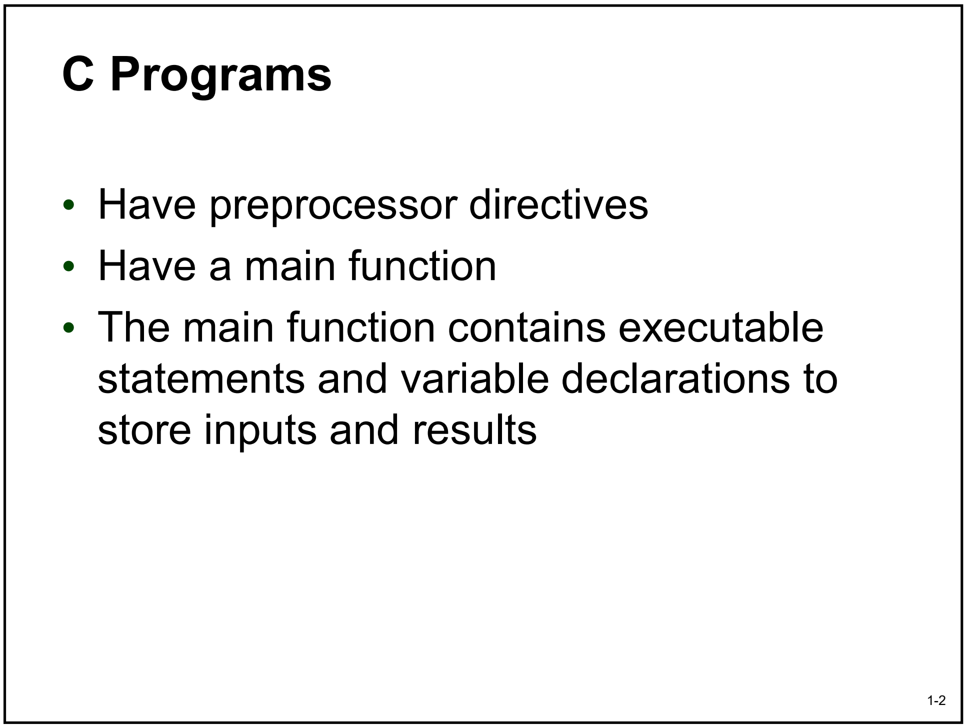
Chapter 2:
Overview of C

Problem Solving & Program Design in C

Seventh Edition

By Jeri R. Hanly &
Elliot B. Koffman

1

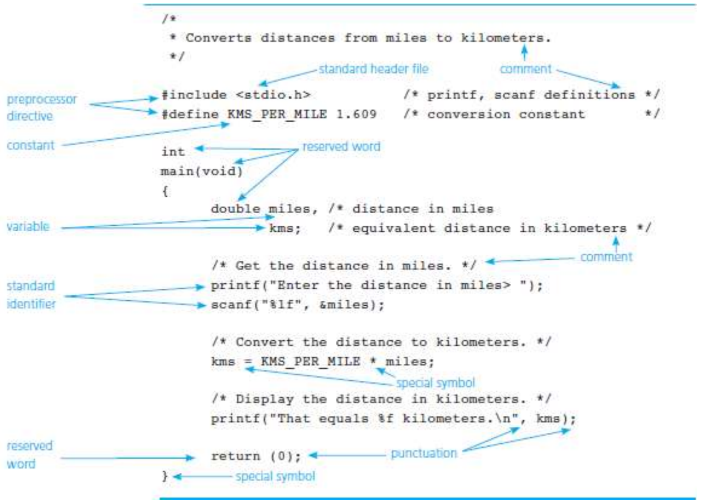


C Programs

- Have preprocessor directives
- Have a main function
- The main function contains executable statements and variable declarations to store inputs and results

2

Figure 2.1 **C Language Elements in Miles-to-Kilometers Conversion Program**

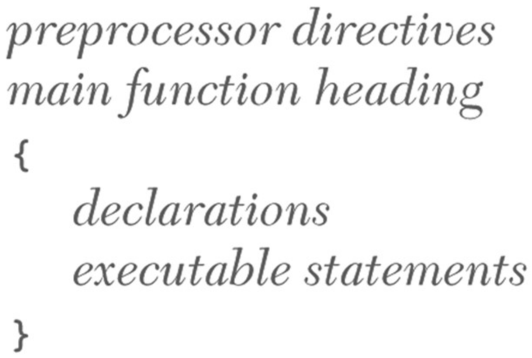


```
/*
 * Converts distances from miles to kilometers.
 */

#include <stdio.h>            /* printf, scanf definitions */
#define KMS_PER_MILE 1.609    /* conversion constant       */

int
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);
}
```

Preprocessor Directives

- #include statements include internal C and programmer defined libraries with useful data and functions
  - #include <stdio.h>
  - #include <math.h>
  - #include "myHeader.h"
- #define statements facilitate the definition of constants and macros
  - #define PI 3.14159
  - #define ADD(a, b) (a + b)

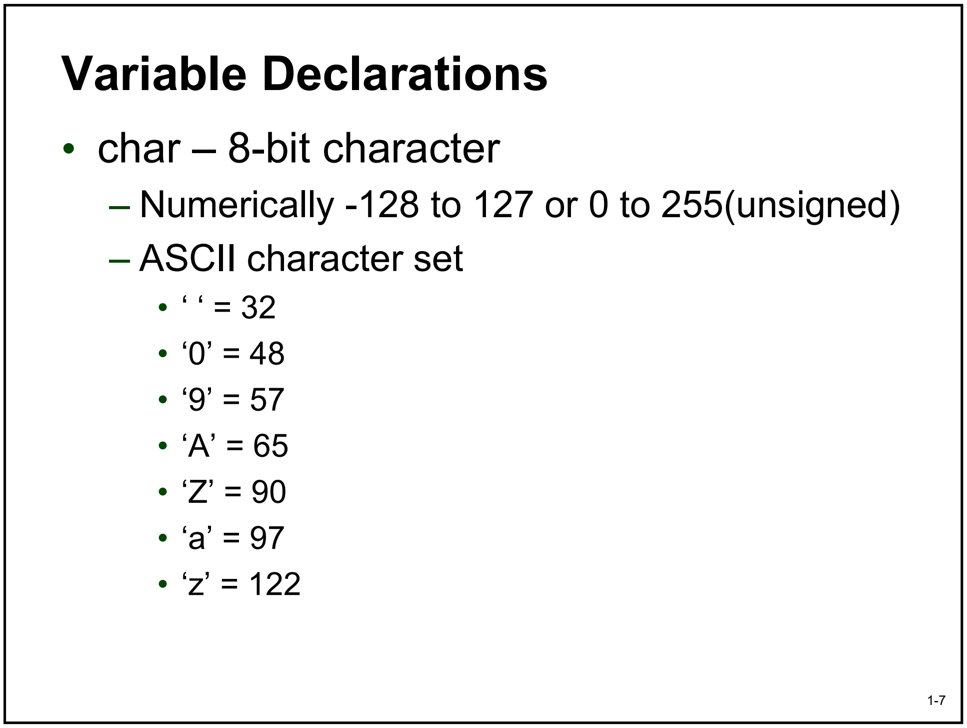
Figure 2.8 **General Form of a C Program**



```
preprocessor directives
main function heading
{
    declarations
    executable statements
}
```

main() function

```
int main(int argc, char *argv[]){

        // Variable declarations

        // Executable statements

        return 0;
}
```

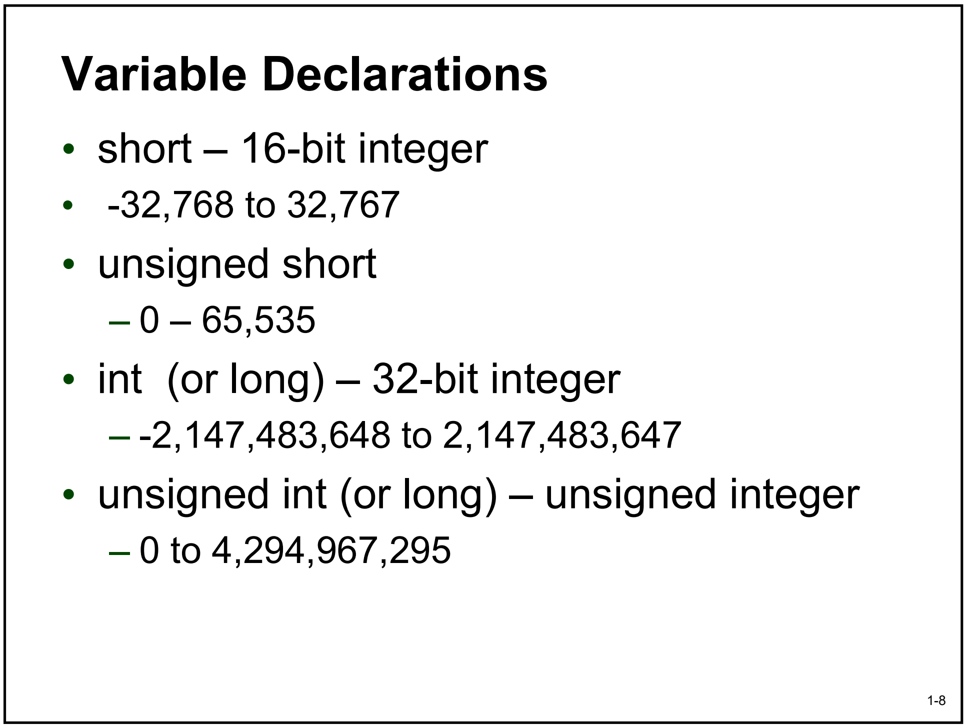


Variable Declarations

- char – 8-bit character
  - Numerically -128 to 127 or 0 to 255(unsigned)
  - ASCII character set
    - ‘ ‘ = 32
    - ‘0’ = 48
    - ‘9’ = 57
    - ‘A’ = 65
    - ‘Z’ = 90
    - ‘a’ = 97
    - ‘z’ = 122

1-7

7

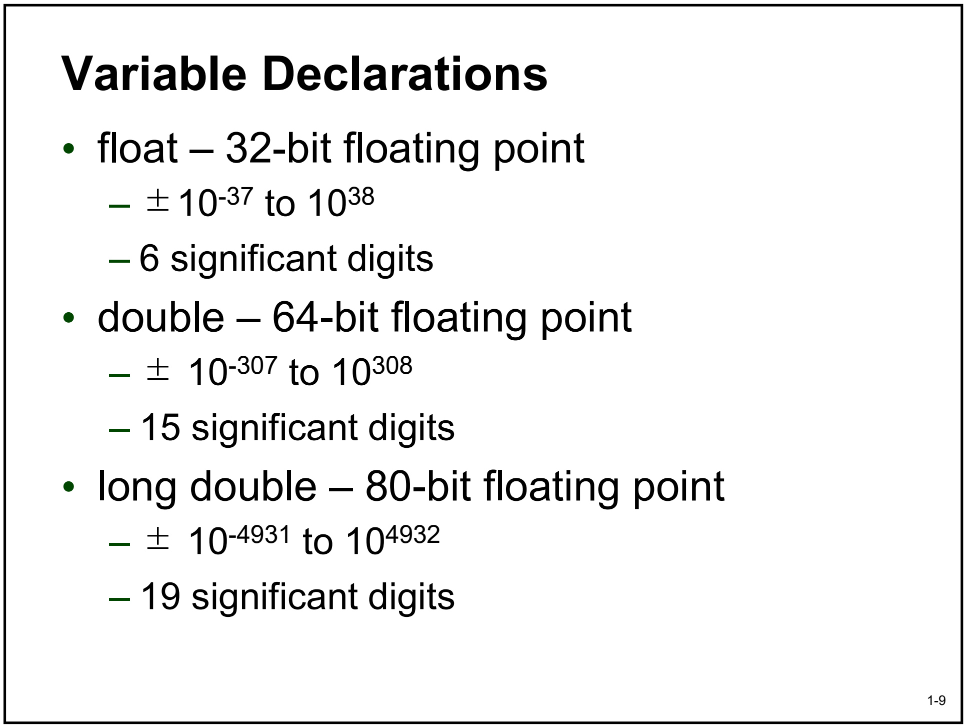


Variable Declarations

- short – 16-bit integer
- -32,768 to 32,767
- unsigned short
  - 0 – 65,535
- int (or long) – 32-bit integer
  - -2,147,483,648 to 2,147,483,647
- unsigned int (or long) – unsigned integer
  - 0 to 4,294,967,295

1-8

8

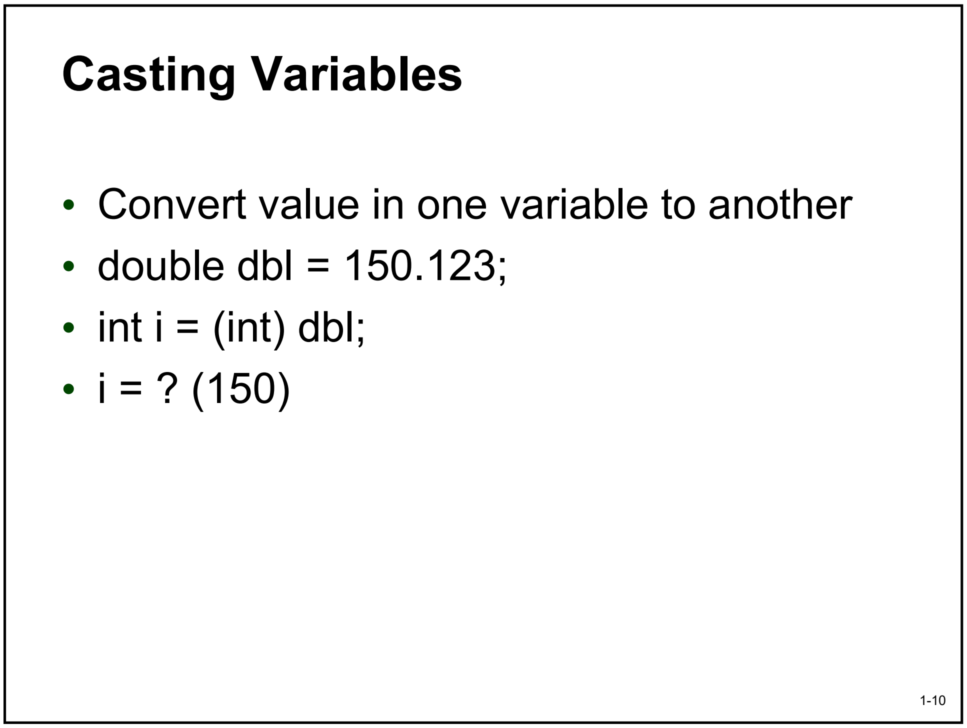


Variable Declarations

- float – 32-bit floating point
  - $\pm 10^{-37}$ to 10^{38}
  - 6 significant digits
- double – 64-bit floating point
  - $\pm 10^{-307}$ to 10^{308}
  - 15 significant digits
- long double – 80-bit floating point
  - $\pm 10^{-4931}$ to 10^{4932}
  - 19 significant digits

1-9

9



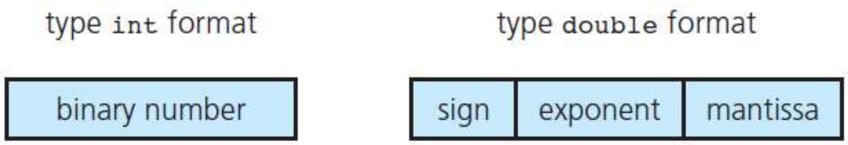
Casting Variables

- Convert value in one variable to another
- double dbl = 150.123;
- int i = (int) dbl;
- i = ? (150)

1-10

10

Figure 2.2 **Internal Format of Type int and Type double**



type `int` format

binary number

type `double` format

sign

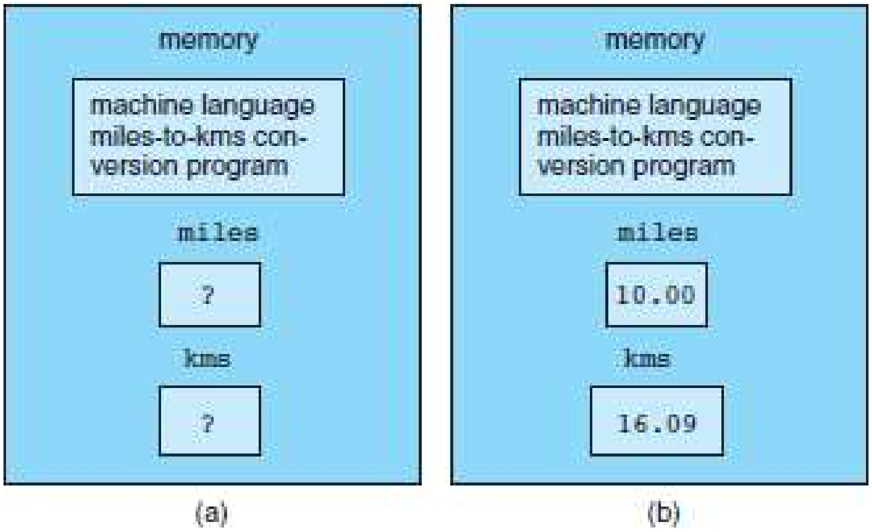
exponent

mantissa

Executable Statements

- printf(“Hello World\n”);
- scanf(“%d”, &variable);
- c = a + b;
- c = pow(a, b);

Figure 2.3 **Memory(a) Before and (b) After Execution of a Program**


memory
machine language miles-to-kms conversion program
miles
?
kms
?
(a)
memory
machine language miles-to-kms conversion program
miles
10.00
kms
16.09
(b)

Assignment

- ‘=‘ is assignment operator
- c = a + b
- b = b + a or b += a

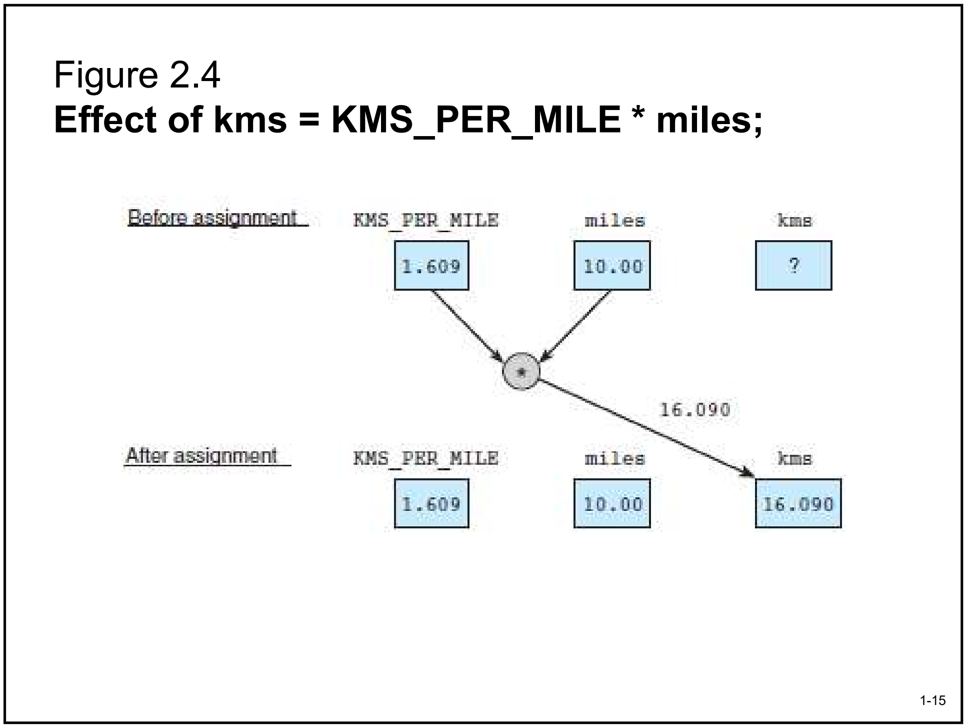


Figure 2.4
Effect of kms = KMS_PER_MILE * miles;

1-15

15

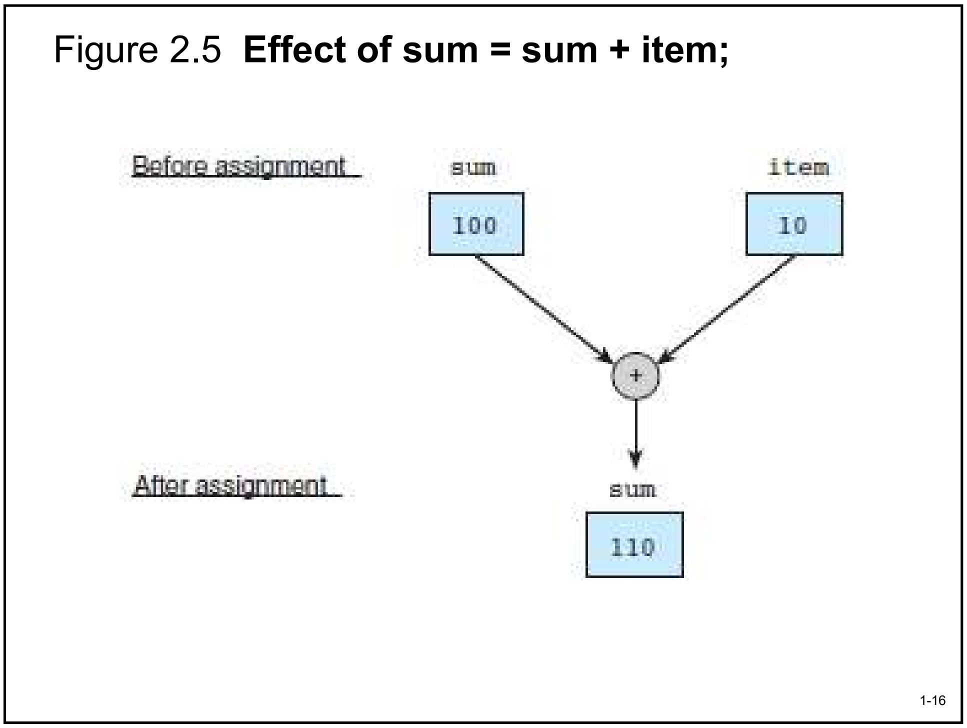
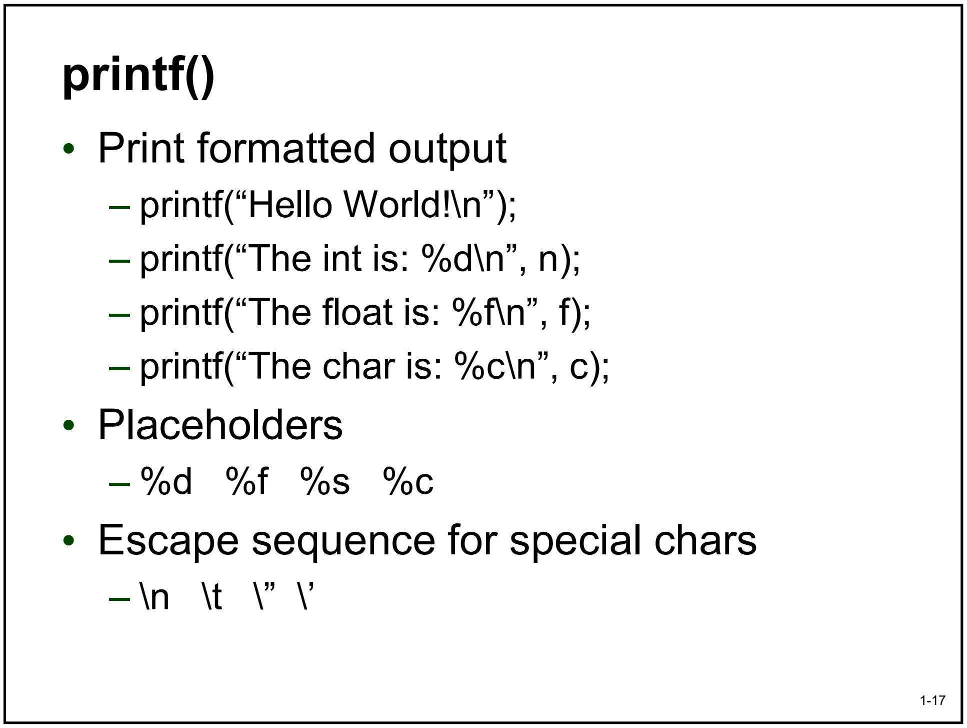


Figure 2.5 **Effect of sum = sum + item;**

1-16

16

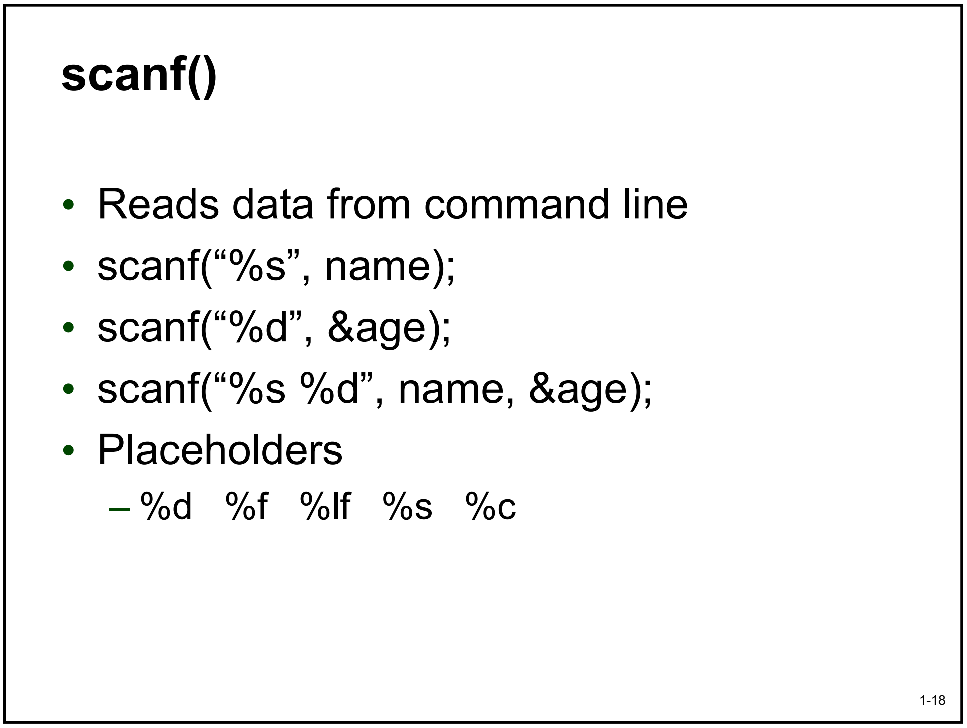


printf()

- Print formatted output
  - printf(“Hello World!\n”);
  - printf(“The int is: %d\n”, n);
  - printf(“The float is: %f\n”, f);
  - printf(“The char is: %c\n”, c);
- Placeholders
  - %d %f %s %c
- Escape sequence for special chars
  - \n \t \” \’

1-17

17



scanf()

- Reads data from command line
- scanf(“%s”, name);
- scanf(“%d”, &age);
- scanf(“%s %d”, name, &age);
- Placeholders
  - %d %f %lf %s %c

1-18

18

Figure 2.6 **Effect of scanf("%lf", &miles);**

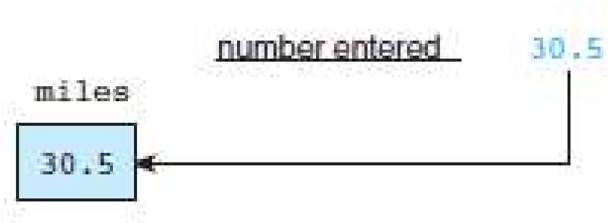
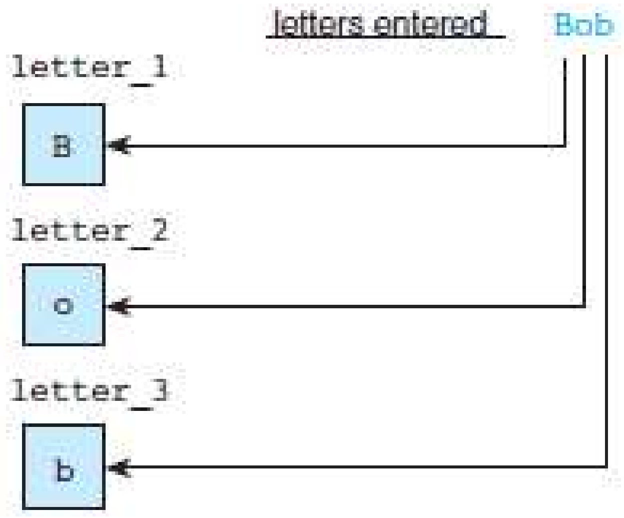


Figure 2.7 **Scanning Data Line Bob**

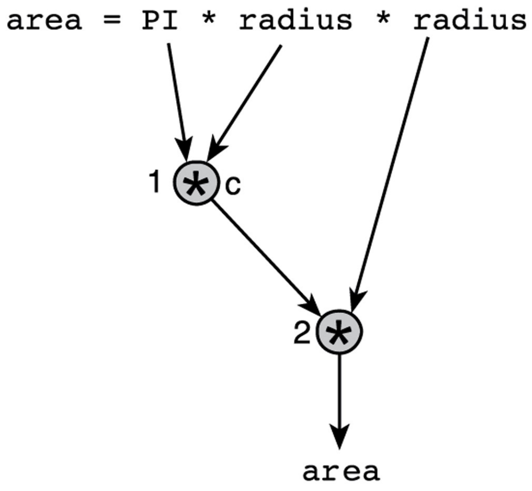


Operator Precedence

```
() [] -> .
! ~ ++ -- + - & * (type)  (unary operators)
* / %
+ -
< <= > >=
== !=
&
|
&&
||
= *= /= %= += -=
```

21

Figure 2.9 **Evaluation Tree for area = PI * radius * radius;**



22

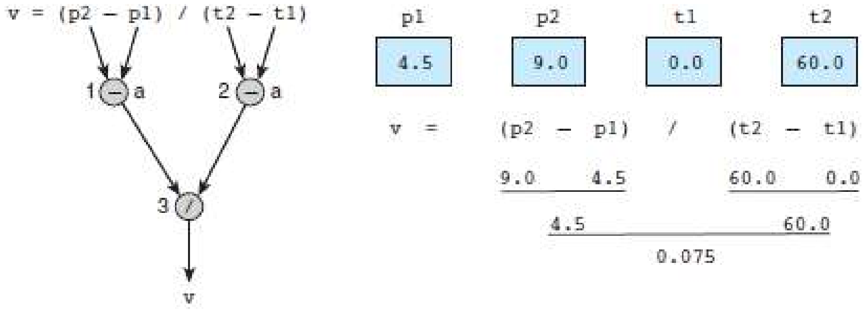
Figure 2.10 **Step-by-Step Expression Evaluation**

```
area    =     PI    *   radius   *   radius
           3.14159         2.0           2.0
           ___________________
                6.28318
                ______________________________
                                12.56636
```

1-23

23

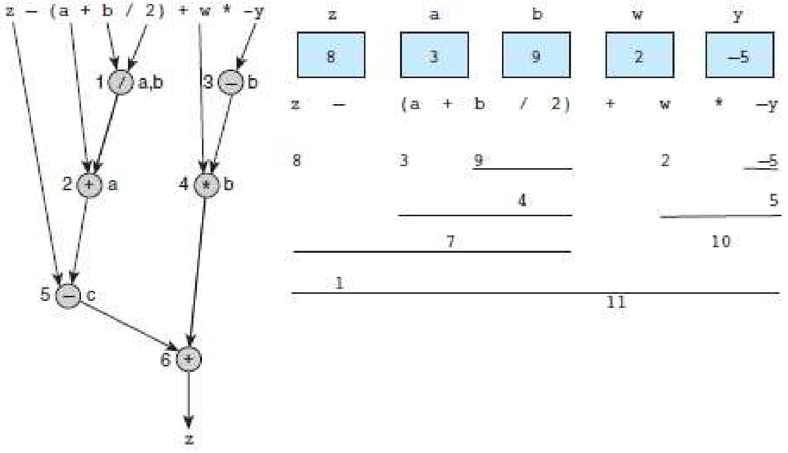
Figure 2.11 **Evaluation Tree and Evaluation for v = (p2 - p1) / (t2 - t1);**



1-24

24

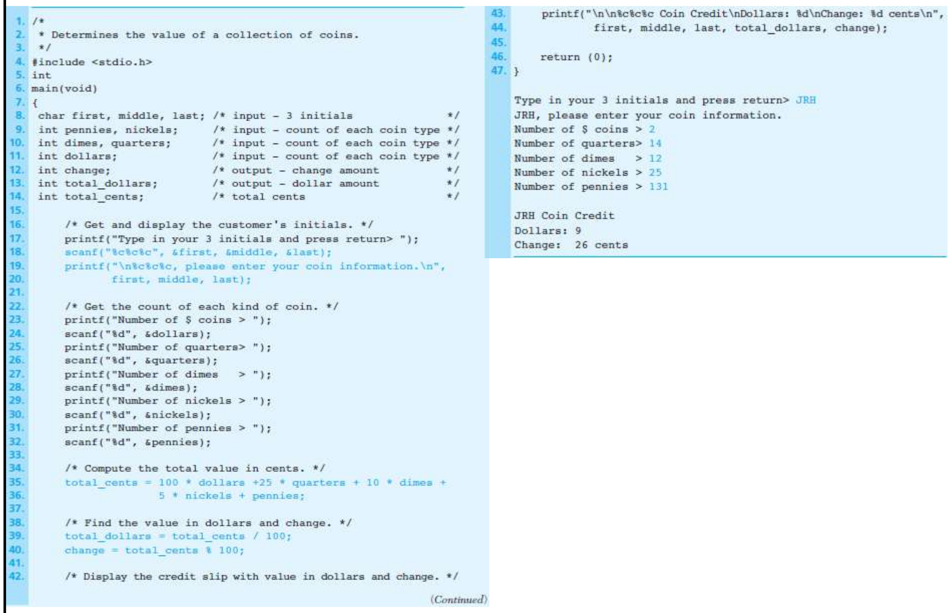
Figure 2.12 **Evaluation Tree and Evaluation for z - (a + b / 2) + w * -y**



Supermarket Coin Machine

- Understand the problem
- Determine data requirements
- Develop algorithm
- Code solution
- Test solution

Figure 2.13 **Supermarket Coin Value Program**



```c
/*
 * Determines the value of a collection of coins.
 */
#include <stdio.h>
int
main(void)
{
 char first, middle, last; /* input - 3 initials              */
 int pennies, nickels;     /* input - count of each coin type */
 int dimes, quarters;      /* input - count of each coin type */
 int dollars;              /* input - count of each coin type */
 int change;               /* output - change amount          */
 int total_dollars;        /* output - dollar amount          */
 int total_cents;          /* total cents                     */

     /* Get and display the customer's initials. */
     printf("Type in your 3 initials and press return> ");
     scanf("%c%c%c", &first, &middle, &last);
     printf("\n%c%c%c, please enter your coin information.\n",
            first, middle, last);

     /* Get the count of each kind of coin. */
     printf("Number of $ coins > ");
     scanf("%d", &dollars);
     printf("Number of quarters> ");
     scanf("%d", &quarters);
     printf("Number of dimes   > ");
     scanf("%d", &dimes);
     printf("Number of nickels > ");
     scanf("%d", &nickels);
     printf("Number of pennies > ");
     scanf("%d", &pennies);

     /* Compute the total value in cents. */
     total_cents = 100 * dollars +25 * quarters + 10 * dimes +
                   5 * nickels + pennies;

     /* Find the value in dollars and change. */
     total_dollars = total_cents / 100;
     change = total_cents % 100;

     /* Display the credit slip with value in dollars and change. */
```

(Continued)

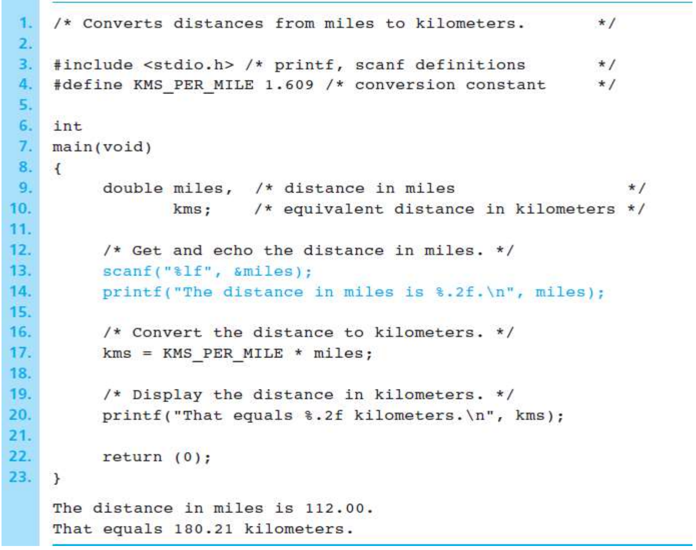
```c
     printf("\n\n%c%c%c Coin Credit\nDollars: %d\nChange: %d cents\n",
            first, middle, last, total_dollars, change);

     return (0);
}
```

```
Type in your 3 initials and press return> JRH
JRH, please enter your coin information.
Number of $ coins > 2
Number of quarters> 14
Number of dimes   > 12
Number of nickels > 25
Number of pennies > 131

JRH Coin Credit
Dollars: 9
Change:  26 cents
```

Batch processing

- Read data from file with scanf
  - executable.exe < infile.txt
- Write data to file with printf
  - executable.exe > outfile.txt

Figure 2.14 **Batch Version of Miles-to-Kilometers Conversion Program**

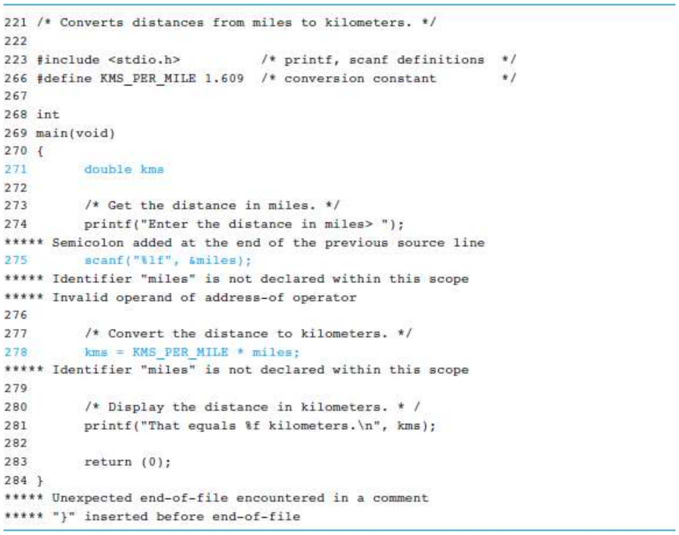


```
/* Converts distances from miles to kilometers.       */

#include <stdio.h> /* printf, scanf definitions       */
#define KMS_PER_MILE 1.609 /* conversion constant     */

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

      /* Get and echo the distance in miles. */
      scanf("%lf", &miles);
      printf("The distance in miles is %.2f.\n", miles);

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

      /* Display the distance in kilometers. */
      printf("That equals %.2f kilometers.\n", kms);

      return (0);
}
The distance in miles is 112.00.
That equals 180.21 kilometers.
```

Errors

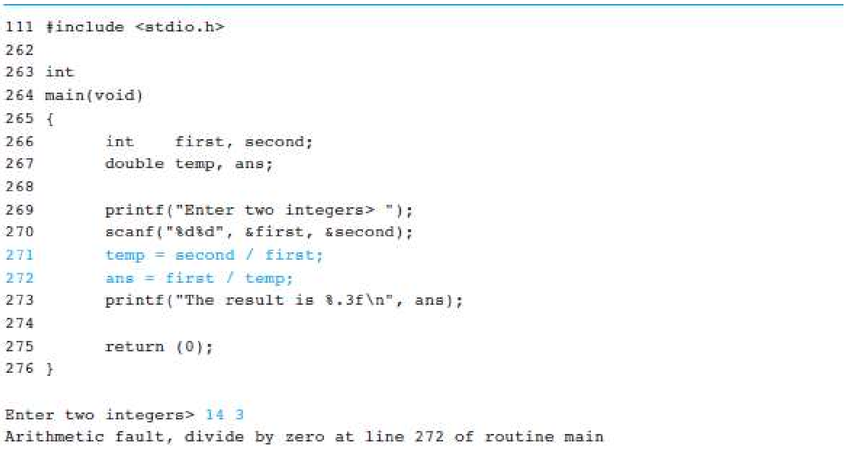
- Syntax or Compile Time Errors
  - Compiler fails to compile and link
- Runtime Errors
  - Program crashes
- Logic Errors
  - Results are incorrect

Figure 2.15 **Compiler Listing of a Program with Syntax Errors**



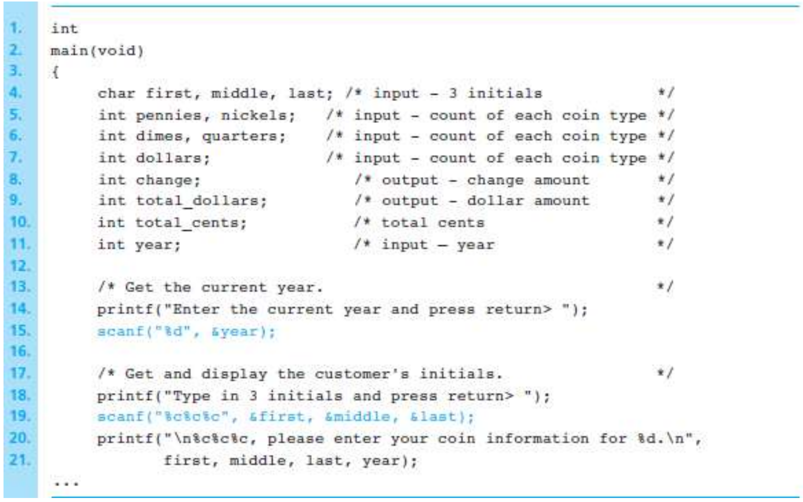
```
221 /* Converts distances from miles to kilometers. */
222
223 #include <stdio.h>          /* printf, scanf definitions  */
266 #define KMS_PER_MILE 1.609  /* conversion constant        */
267
268 int
269 main(void)
270 {
271       double kms
272
273       /* Get the distance in miles. */
274       printf("Enter the distance in miles> ");
***** Semicolon added at the end of the previous source line
275       scanf("%lf", &miles);
***** Identifier "miles" is not declared within this scope
***** Invalid operand of address-of operator
276
277       /* Convert the distance to kilometers. */
278       kms = KMS_PER_MILE * miles;
***** Identifier "miles" is not declared within this scope
279
280       /* Display the distance in kilometers. * /
281       printf("That equals %f kilometers.\n", kms);
282
283       return (0);
284 }
***** Unexpected end-of-file encountered in a comment
***** "}" inserted before end-of-file
```

Figure 2.16 **A Program with a Run-Time Error**



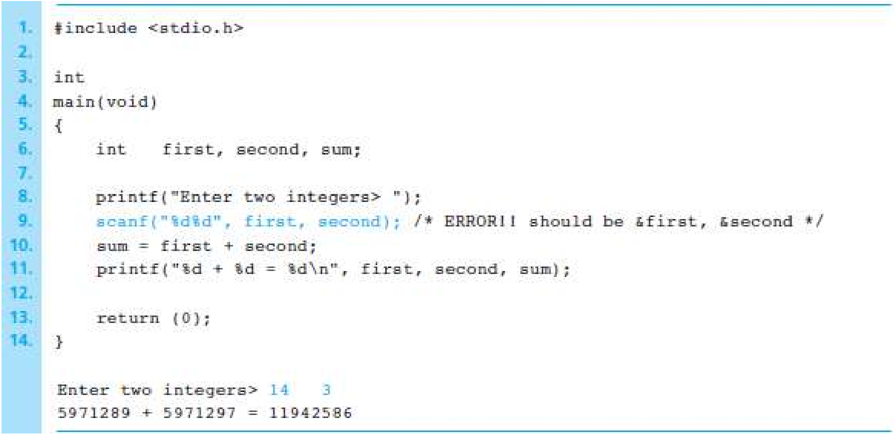
```
111 #include <stdio.h>
262
263 int
264 main(void)
265 {
266       int    first, second;
267       double temp, ans;
268
269       printf("Enter two integers> ");
270       scanf("%d%d", &first, &second);
271       temp = second / first;
272       ans = first / temp;
273       printf("The result is %.3f\n", ans);
274
275       return (0);
276 }

Enter two integers> 14 3
Arithmetic fault, divide by zero at line 272 of routine main
```

Figure 2.17 **Revised Start of main Function for Supermarket Coin Value Program**



```
int
main(void)
{
      char first, middle, last; /* input - 3 initials            */
      int pennies, nickels;   /* input - count of each coin type */
      int dimes, quarters;    /* input - count of each coin type */
      int dollars;            /* input - count of each coin type */
      int change;                /* output - change amount       */
      int total_dollars;         /* output - dollar amount       */
      int total_cents;           /* total cents                  */
      int year;                  /* input — year                 */

      /* Get the current year.                                   */
      printf("Enter the current year and press return> ");
      scanf("%d", &year);

      /* Get and display the customer's initials.                */
      printf("Type in 3 initials and press return> ");
      scanf("%c%c%c", &first, &middle, &last);
      printf("\n%c%c%c, please enter your coin information for %d.\n",
             first, middle, last, year);
...
```

Figure 2.18 **A Program That Produces Incorrect Results Due to & Omission**



```
#include <stdio.h>

int
main(void)
{
    int   first, second, sum;

    printf("Enter two integers> ");
    scanf("%d%d", first, second); /* ERROR!! should be &first, &second */
    sum = first + second;
    printf("%d + %d = %d\n", first, second, sum);

    return (0);
}

Enter two integers> 14   3
5971289 + 5971297 = 11942586
```