Chapter 2 Overview of C

Section 2.1

2. E should be defined as a constant macro because its value should not change during program execution. If for some reason the value would need to be globally changed, using a constant macro confines this change to one place.

4. You shouldn't use a standard identifier as the name of a memory cell because by doing so you lose the ability to use that library function in your program. No, you cannot use a reserved word instead.

Section 2.2

2.

Constants	Valid/Invalid	Data Type
'PQR'	invalid	
15E-2	valid	double
35	valid	int
'h'	valid	char
-37.491	valid	double
.912	valid	double
4,719	invalid	
'true'	invalid	
"T"	not one of the listed types	
&	invalid	
4.5e3	valid	double
'$'	valid	char

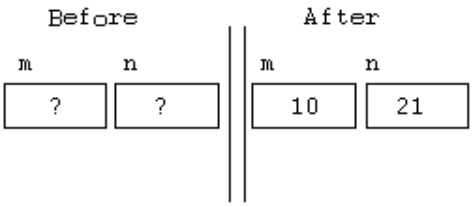
Programming

1.
```
#include <stdio.h>
#define PI 3.14159

int
main(void)
{
  double  radius, area, circumf;
  int     num_circ;
  char    circ_name;
  /* executable statements omitted */
}
```

Section 2.3

2.



4. One option is to delete the third line,
printf("\n");
and insert a space after the period on the second line,
printf("Jane Doe. ");

Programming

1.
```
int first, second, third;

printf("\nEnter three integers> ");
scanf("%d%d%d", &first, &second, &third);
```

2.
```
a.    printf("The value of n is %d.\n", n);
b.    printf("The area of a square whose side length is %f cm is %f\n square cm.\n",
                side, area);
```

3.
```
#include <stdio.h>
#define PI 3.14159

int
```

```
main(void)
{
   double radius, area;

   /* Prompt for user input and get the radius. */
   printf("Enter the radius> ");
   scanf("%lf", &radius);

   /* Compute the area. */
   area = PI * radius * radius;

   /* Display the results. */
   printf("\nThe area is %f\n", area);

   return (0);
}
```

Section 2.4

2.
```
/*
 * Calculate and display the difference of two input values
 */

#include <stdio.h>

int
main(void)
{
   int first_num,         /* first input value */
   second_num,            /* second input value */
   sum;                   /* sum of inputs */

   scanf("%d%d", &first_num, &second_num);
   sum = first_num + second_num;
   printf("%d + %d = %d\n", first_num, second_num, sum);

   return (0);
}
```

The first statement scans two integers, first_num and second_num. (It would be better to include a printf statement first displaying a prompting message.) The second statement sums the two integers entered. The third statement displays the first number, a plus sign, the second number, an equal sign, and the sum of the first and second numbers.

Programming

1.
```
#include <stdio.h>

int
main(void)
{
   char let;
   double num;

   /* Prompt for user input and scan. */
   printf("\nEnter a character> ");
   scanf("%c", &let);
   printf("\nEnter a number> ");
   scanf("%lf", &num);

   /* Display the results. */
   printf("\nThe character is '%c'.\nThe number is %.2f.\n", let, num);

   return (0);
}
```

Section 2.5

2.
```
1.8   *   celsius   +   32.0
                38.1
      68.58
                  100.58
```

```
(salary    -    5000.00)    *    0.20    +    1425.00
38450.00
       33450.00
                          6690.00
                                                8115.00
```

4.

a. 1	j. 1
b. ??	k. 3.5
c. 3	l. 21
d. 6.28318	m. 1.570795
e. ??	n. 0.0
f. 2.0	o. 3
g. 1.0	p. undefined
h. undefined	q. 7
i. ??	r. 2.3333333…

(?? means the result varies)

6.
```
a. a = a % c
b. x = (3 * a) / (b * c)
c. j = 4 * (i + k);
```

8. Representational error occurs when the number of bits (binary digits) in the mantissa of a type double variable is insufficient to exactly represent a certain fraction. Cancellation error occurs when performing an operation on two numbers that have a very large difference in magnitude, and the smaller number's effect is lost.

10. x is 10.5, y is 7.2, m is 5, n is 2

a. x / (double)m is 2.1

b. x / m is 2.1

c. (double)(n * m) is 10.0

d. (double)(n / m) + y is 7.2

e. (double)(n / m) is 0.0

Programming

1.
```
q = (k * A * (T1 - T2)) / L;
```

2. One option:
```
#include <stdio.h>

#define LETTER 'A'

int
main(void)
{
   char letter1, letter2;
   int val1;
   double val2;

   letter1 = LETTER;

   /* Prompt for user input and scan data. */
   printf("\nEnter one character, one integer, and another number> ");
   scanf("%c%d%lf", &letter2, &val1, &val2);
      …

   return (0);
}
```

Section 2.6

2. value = -3.6175 (# means blank)

Format	Output
%8.4f	#-3.6175
%8.3f	##-3.618
%8.2f	###-3.62
%8.1f	####-3.6
%8.0f	#####-4.
%.2f	-3.62

Programming

1. `printf("%5d%11.2f%9.1f", a, b, c);`

Section 2.7

2. In an interactive program the data is taken from keyboard input. In a batch program the input comes from a file.

Programming

1.
```
/*
 * Gets three input characters which are user's initials and displays
 * them in a welcoming message.  Then gets input of the quantities of
 * each of the following coins, in the respective order, quarters,
 * dimes, nickels, and pennies.  Computes the total value of the
 * coins, and then displays the value.  Input is taken from a file
 * provided through input redirection.  Output can be redirected to
 * a file if desired.
 */

#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 change;               /* output - change amount          */
   int dollars;              /* output - dollar amount          */
   int total_cents;          /* total cents                     */

   /* Get and display the customer's initials. */
   scanf("%c%c%c", &first, &middle, &last);
   printf("\nHello %c%c%c, let's see what your coins are worth.\n",
           first, second, third);

   /* Get the count of each kind of coin. */
   scanf("%d", &quarters);
   printf("Number of quarters is %1d.\n", quarters);
   scanf("%d", &dimes);
   printf("Number of dimes is %1d.\n", dimes);
   scanf("%d", &nickels);
   printf("Number of nickels is %1d.\n", nickels);
   scanf("%d", &pennies);
   printf("Number of pennies is %1d.\n", pennies);

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

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

   /* Display the value in dollars and change. */
   printf("\nYour coins are worth %d dollars and %d cents.\n",
           dollars, change);*/

   return (0);
}
```