

Chapter 1

Computer Science and Programming

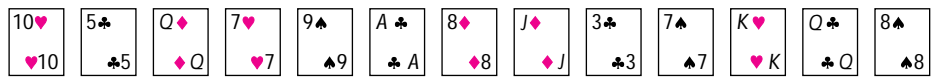


Figure 1.1:

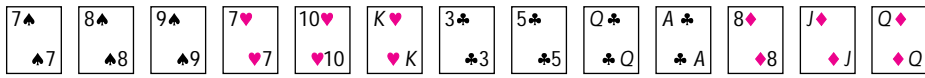


Figure 1.2:

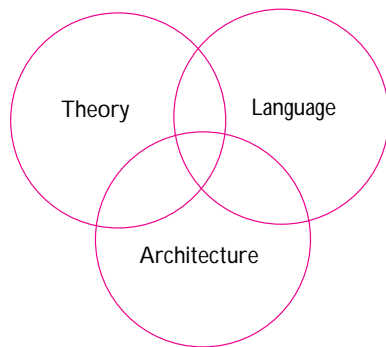


Figure 1.3:

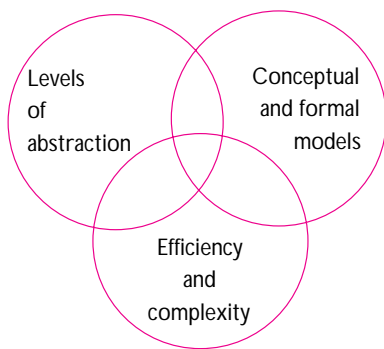


Figure 1.4:

Rational	Integer	Real	Complex
$3/4 * 8/9$	$1,285 * 57$	$3.14 * 6.023$	$(3 + 5i) * (2 - 7i)$
$2/3$	$73,245$	18.91222	$41 - 11i$

Figure 1.5:

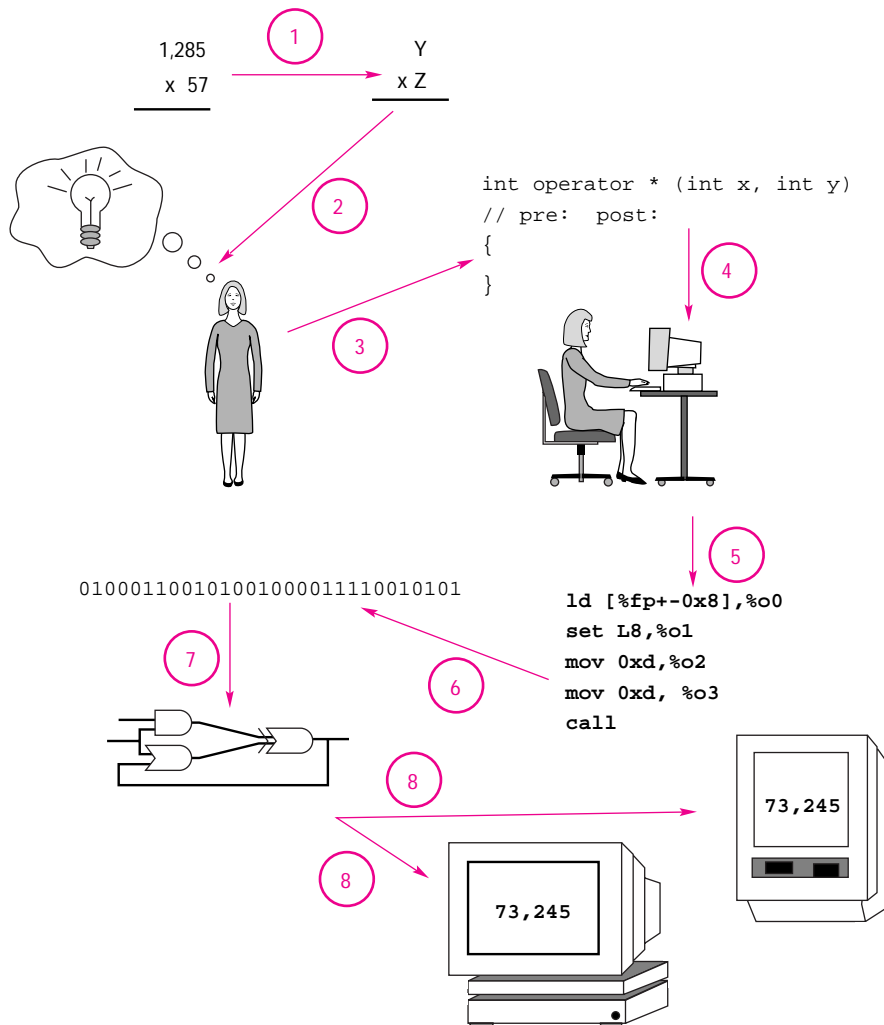


Figure 1.6:

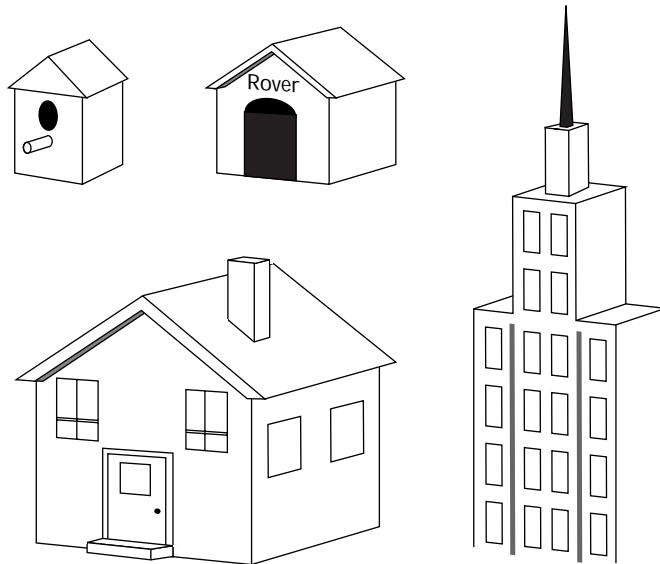


Figure 1.7:



Figure 1.8:

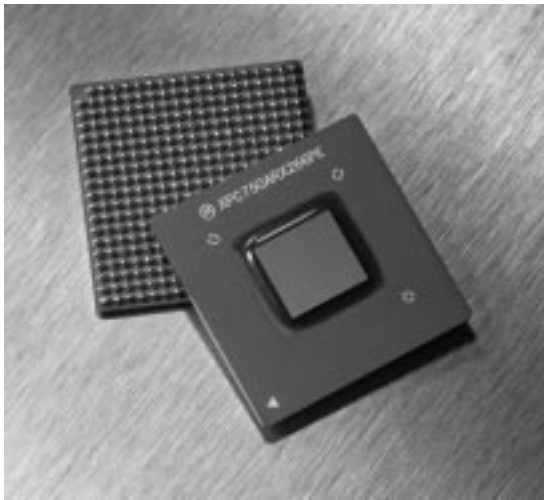


Figure 1.9:

Chapter 2

C++ Programs: Form and Function

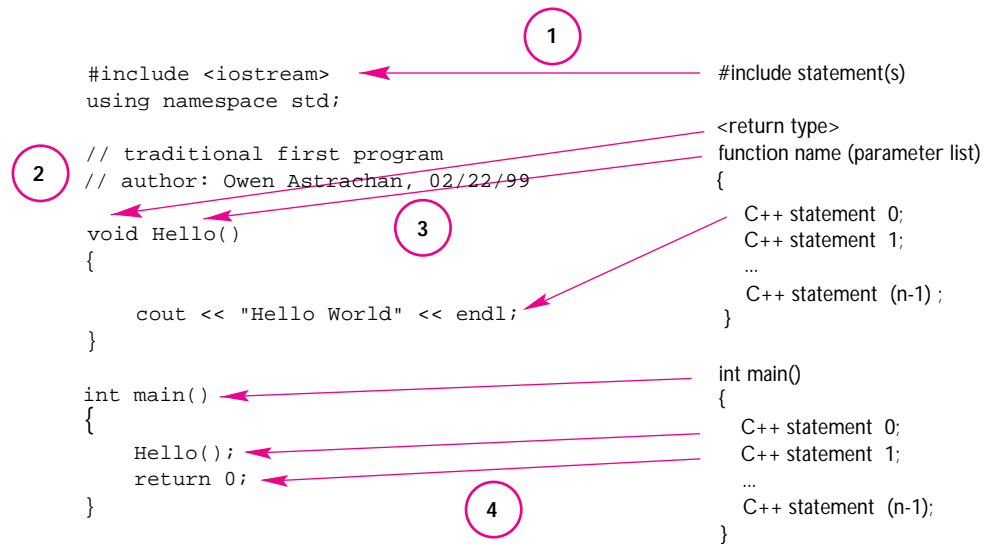


Figure 2.1:

```
void Sing(string person)
{
    "Grace"
    cout << "Happy birthday to you" << endl;
    cout << "Happy birthday to you" << endl;
    cout << "Happy birthday dear " << person << endl;
    cout << "Happy birthday to you" << endl;
    cout << endl;
}

int main()
{
    Sing("Grace");
    Sing("Alan");
    ...
}
```

Figure 2.2:

```
void HadA(string animal)
{
    "pig"
    cout << "and on his farm he had a " << animal << ", ";
    EiEiO();
}

void WithA(string noise)
{
    "oink"
    cout << "With a " << noise << " " << noise << " here" << endl;
    ...
}

void Pig()
{
    Refrain();
    HadA("pig");
    WithA("oink");
    Refrain();
}
```

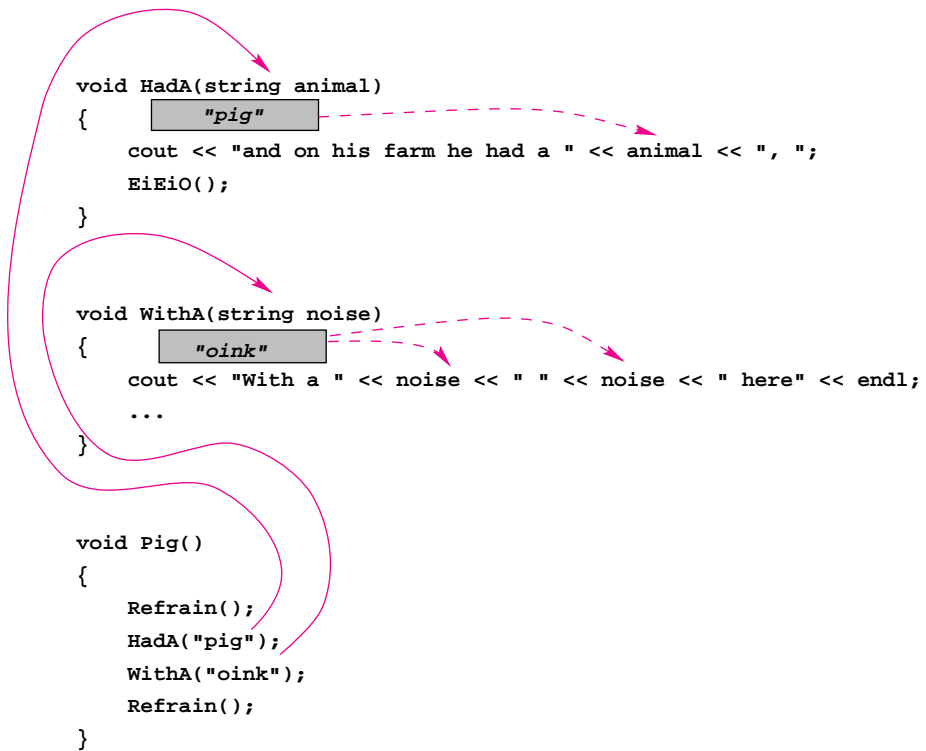


Figure 2.3:

```
void
Verse(string animal, string noise)
{
    "pig" "oink"
    Refrain()
    Had(animal);
    WithA(noise);
    Refrain();
}

int main()
{
    Verse("pig","oink");
    ...
}
```

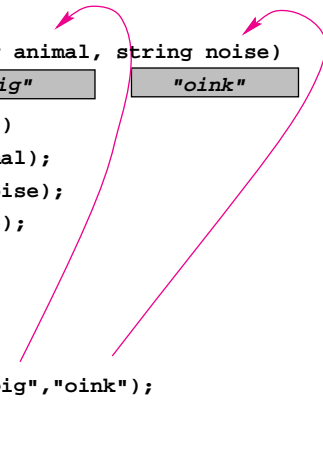


Figure 2.4:

Chapter 3

Program Design and Implementation

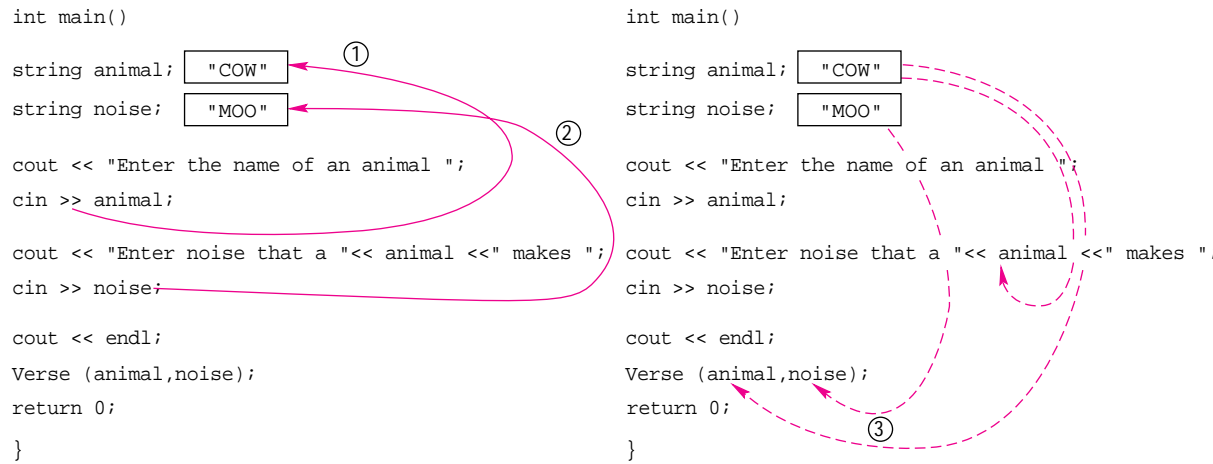


Figure 3.1:

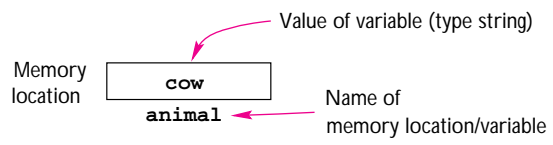


Figure 3.2:

```
      28 ← 1347/47
47 | 1347
   | 1316
   | 31 ← 1347 % 47
```

Figure 3.3:

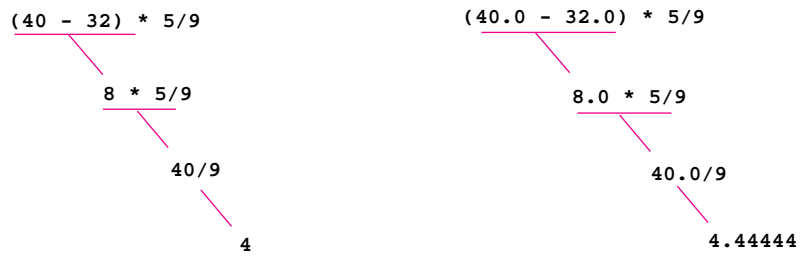


Figure 3.4:

```
void SlicePrice(int radius, double price)
{
    ...
    ...
    cout << "$" < 3.14159*radius*radius/price
    ...
}

int main()
{
    int radius;
    double price;
    ...
    SlicePrice(radius,price);
}
```

Figure 3.5:

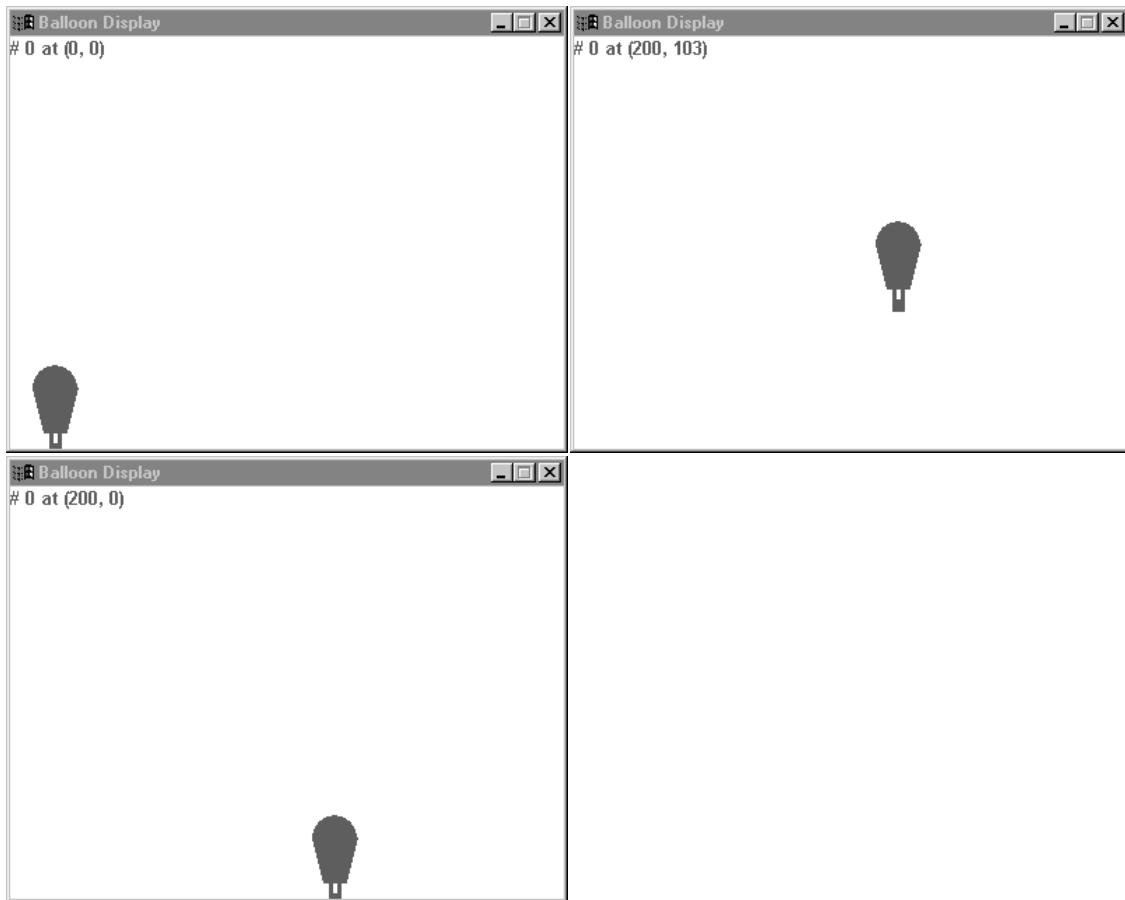


Figure 3.6:

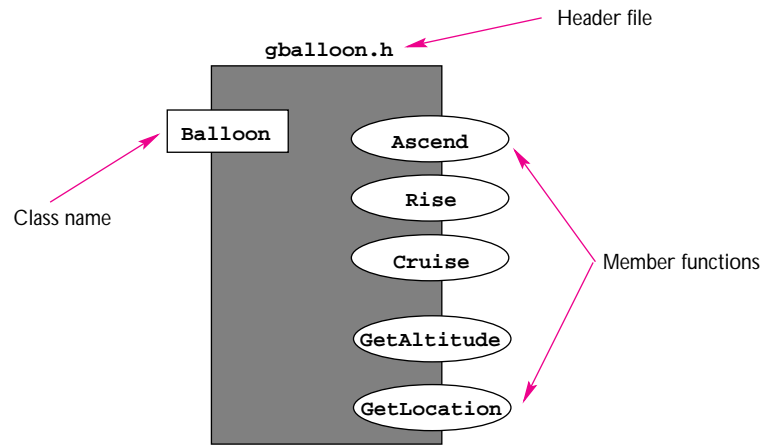


Figure 3.7:

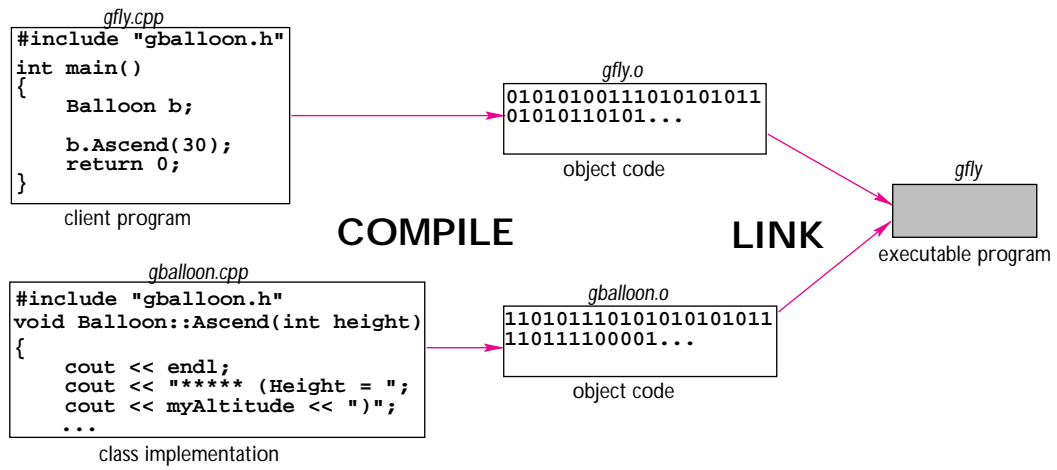


Figure 3.8:

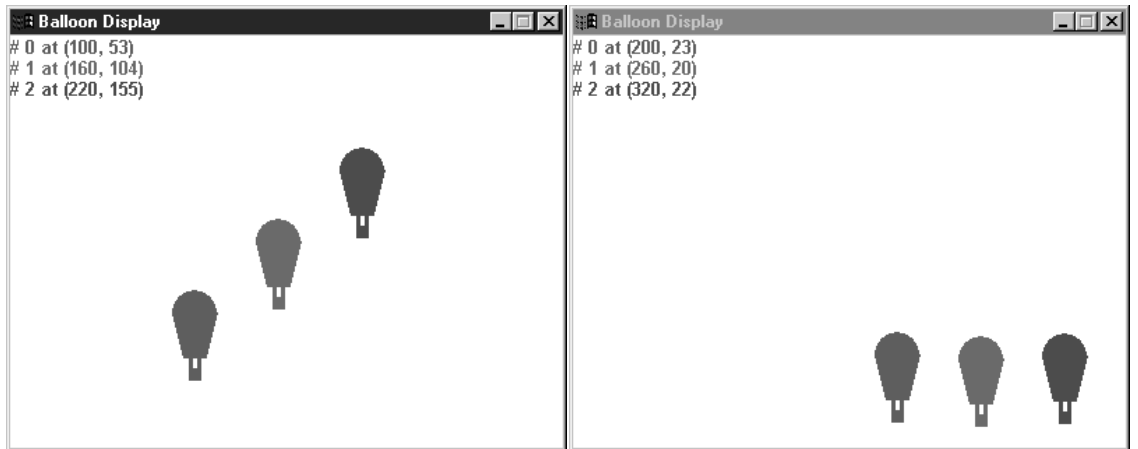


Figure 3.9:

Chapter 4

Control, Functions, and Classes

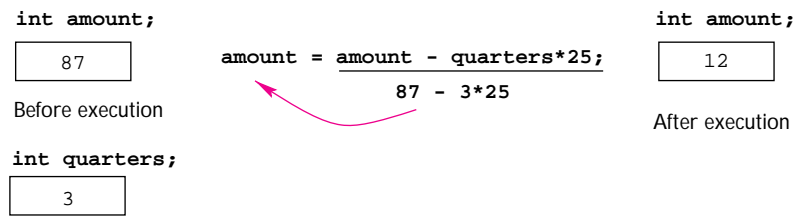


Figure 4.1:

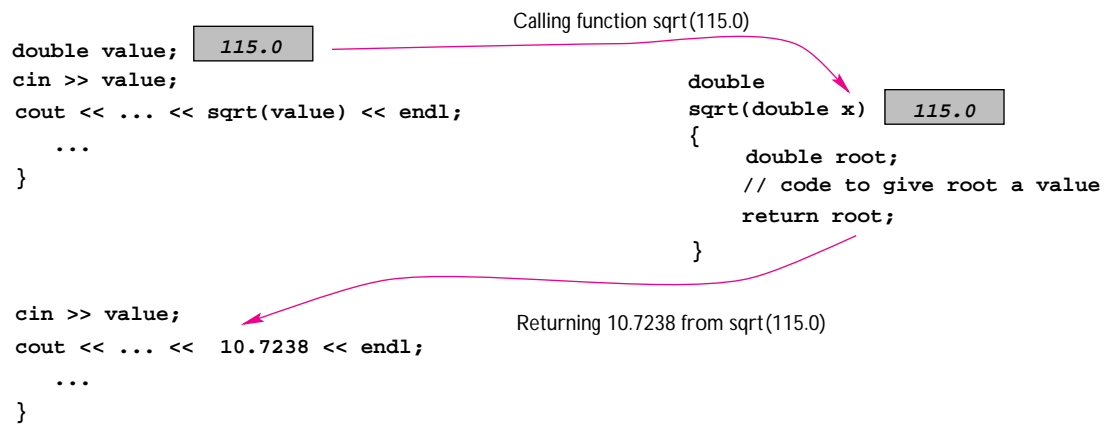


Figure 4.2:

Chapter 5

Iteration with Programs and Classes

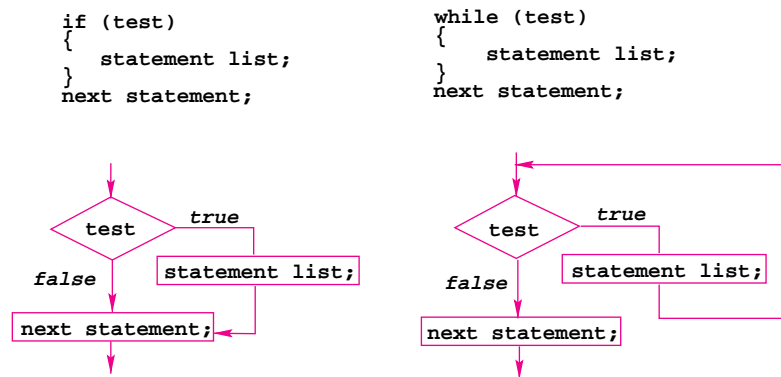


Figure 5.1:

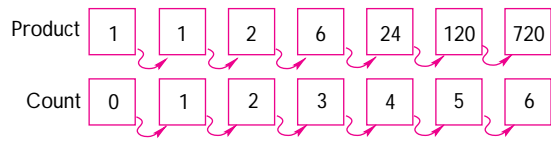


Figure 5.2:

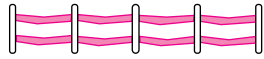


Figure 5.3:

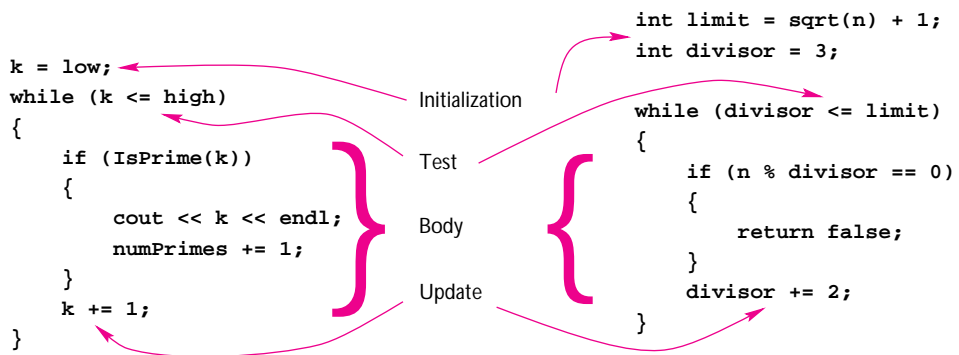


Figure 5.4:

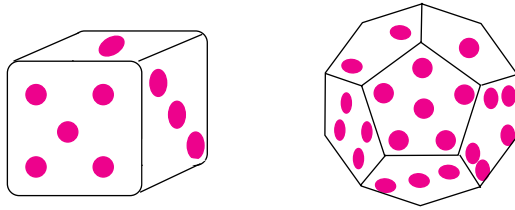


Figure 5.5:

Chapter 6

Classes, Iterators, and Patterns

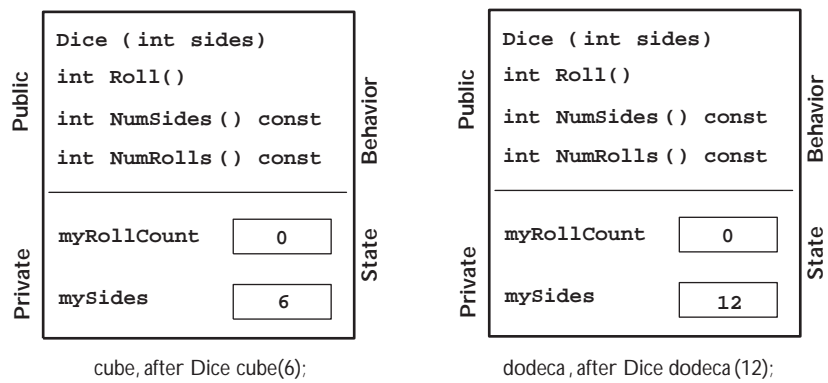


Figure 6.1:

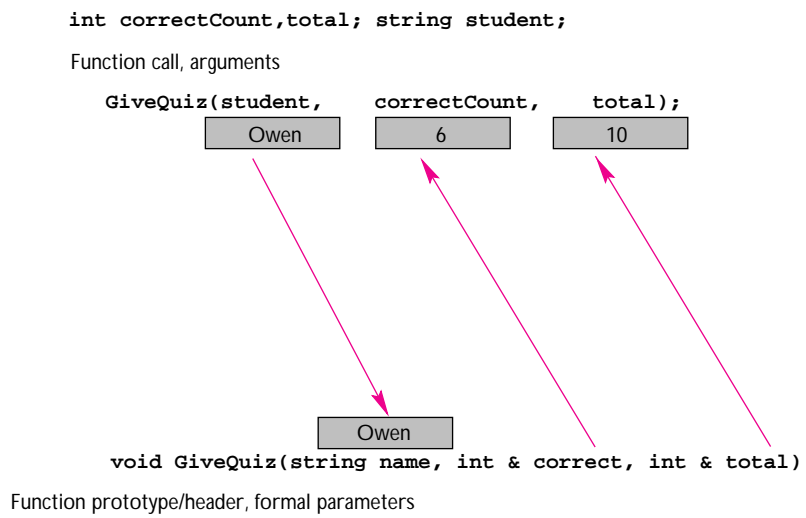


Figure 6.2:

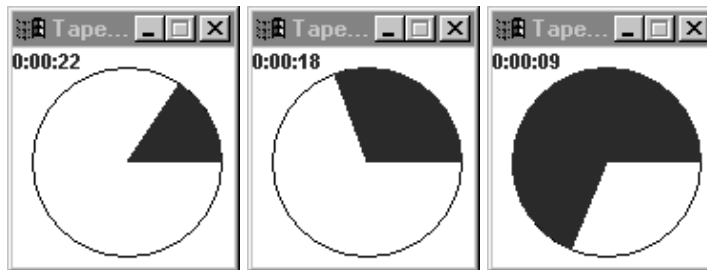


Figure 6.3:

Chapter 7

Class Interfaces, Design, and Implementation

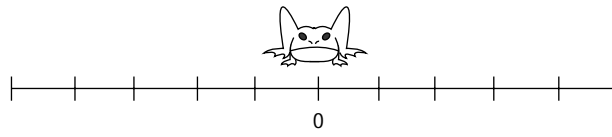


Figure 7.1:

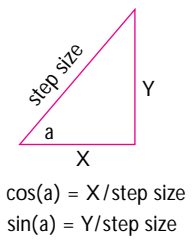


Figure 7.2:

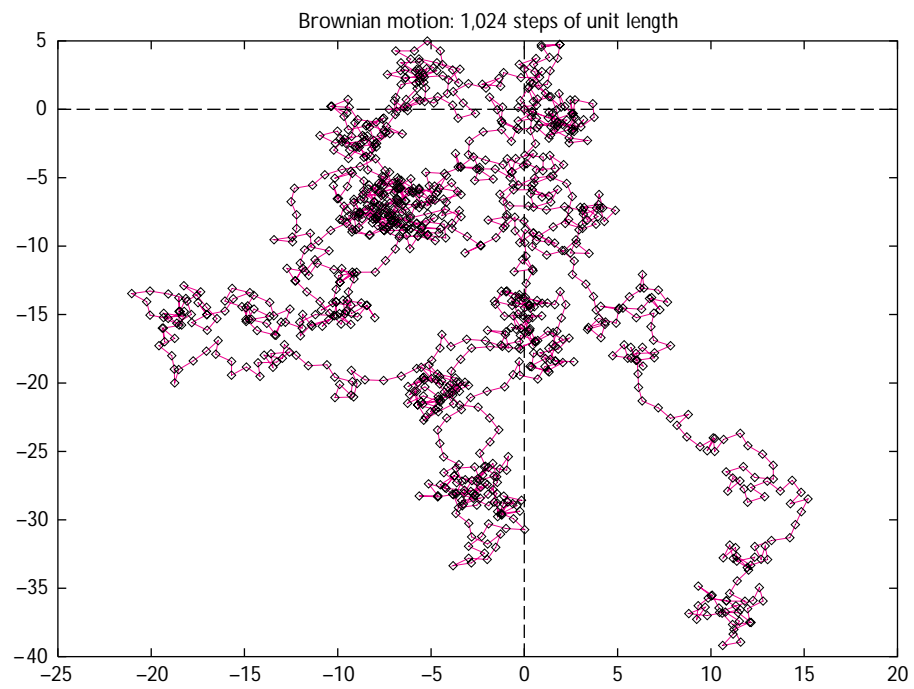


Figure 7.3:

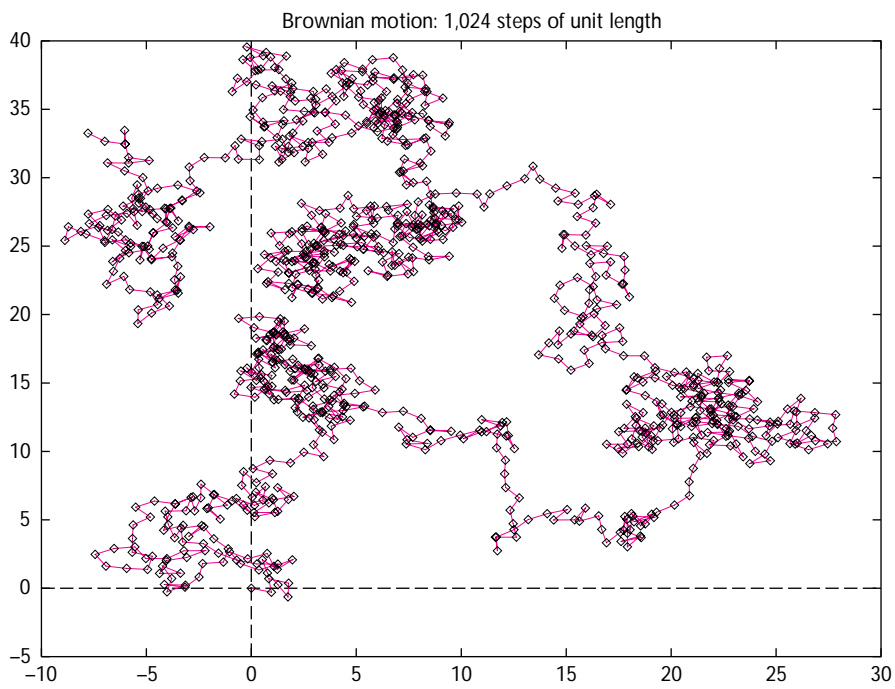


Figure 7.4:

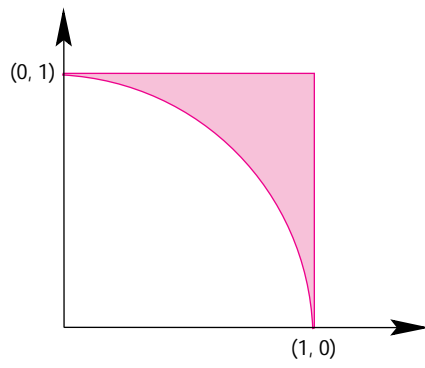


Figure 7.5:

Chapter 8

Arrays, Data, and Random Access

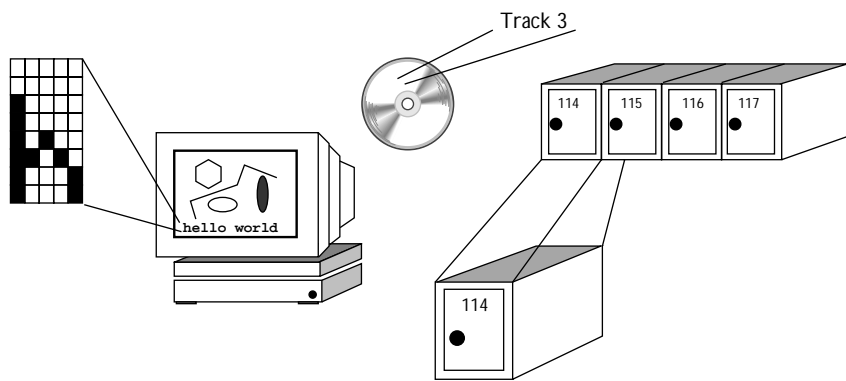


Figure 8.1:

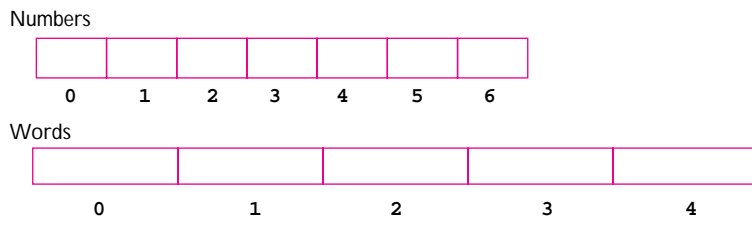


Figure 8.2:

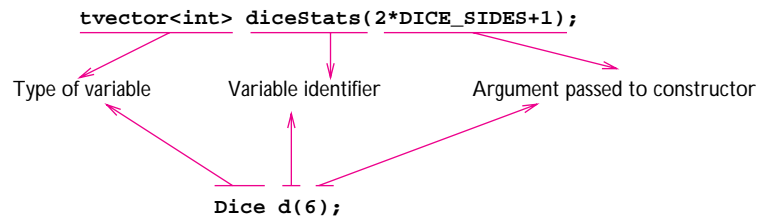


Figure 8.3:

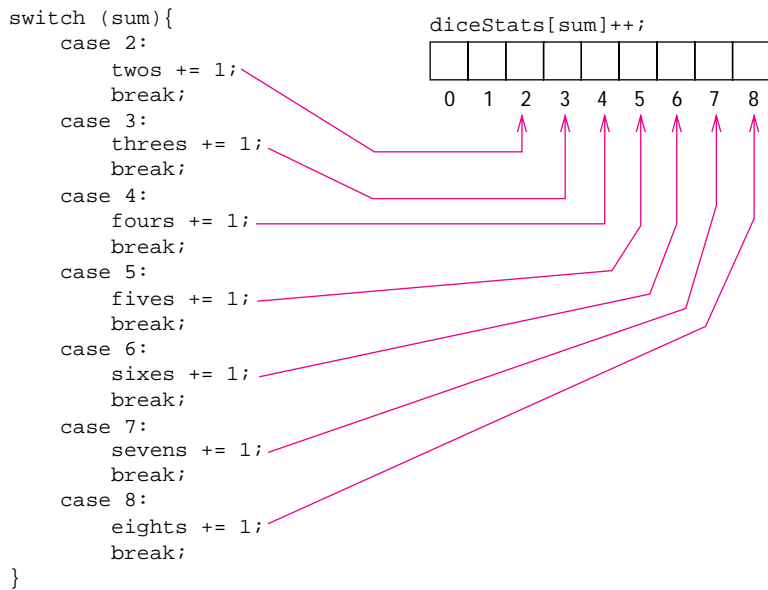
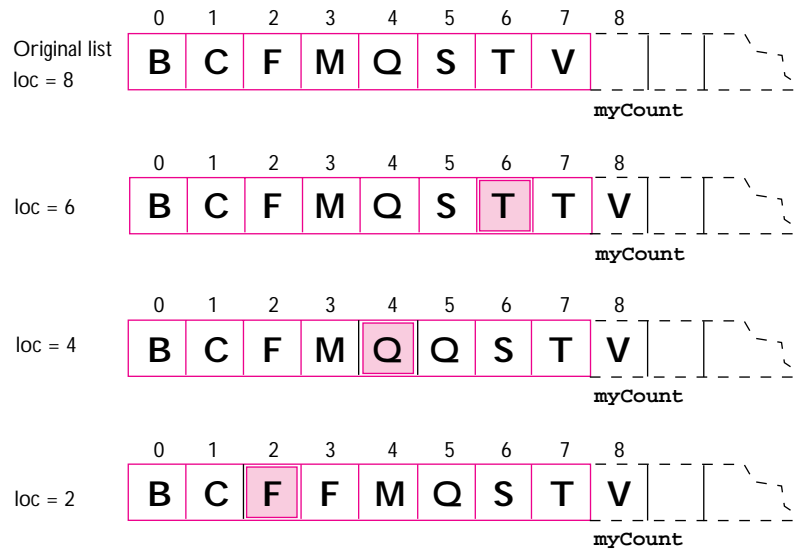


Figure 8.4:



Add **D** to vector maintained in sorted order

Figure 8.5:

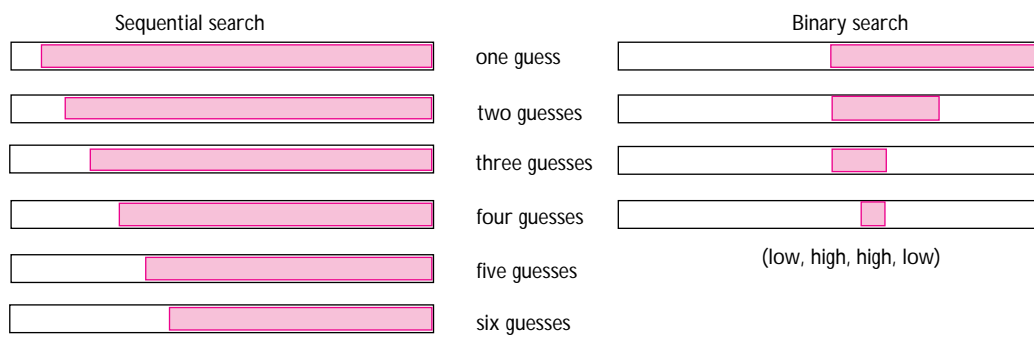


Figure 8.6:



Figure 8.7:

Chapter 9

Strings, Streams, and Operators

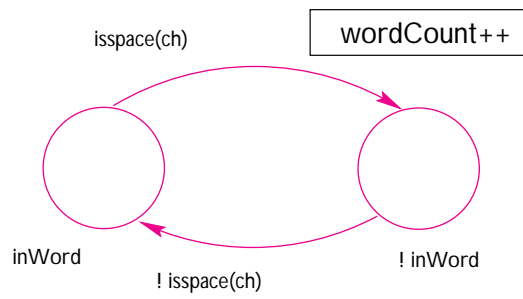


Figure 9.1:

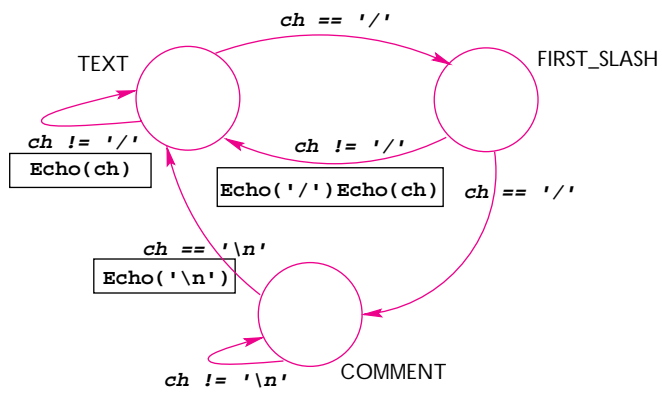


Figure 9.2:

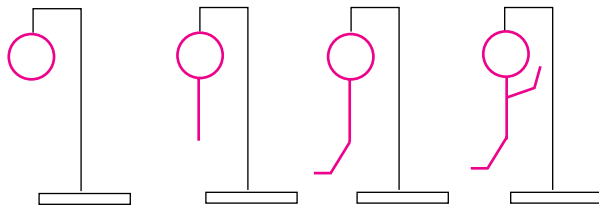


Figure 9.3:

Chapter 10

Recursion, Lists, and Matrices

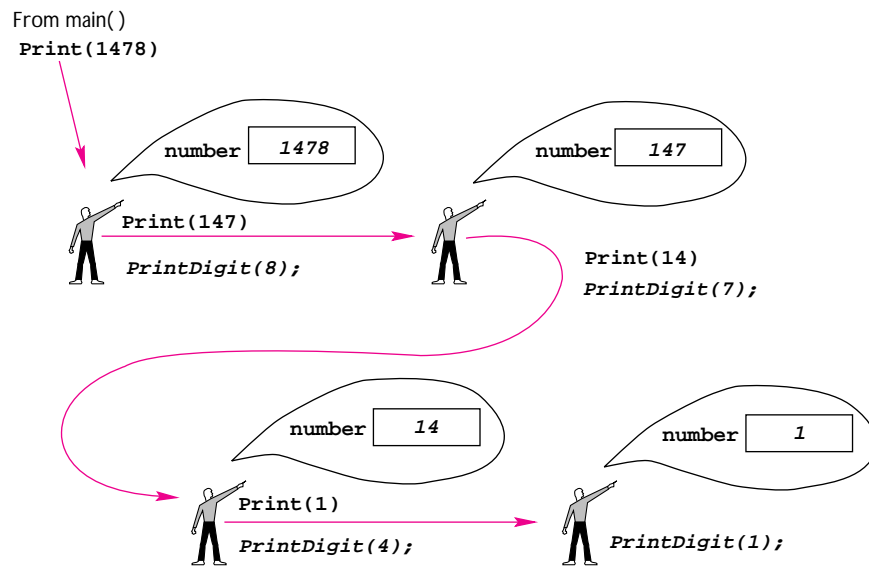


Figure 10.1:

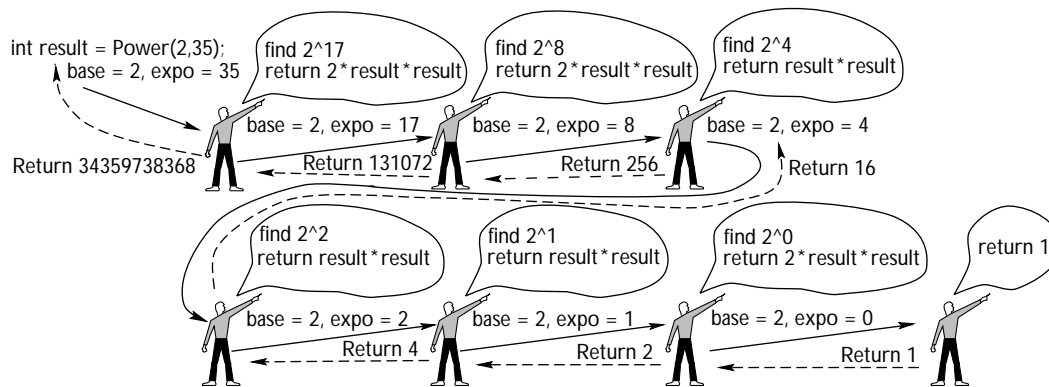


Figure 10.2:

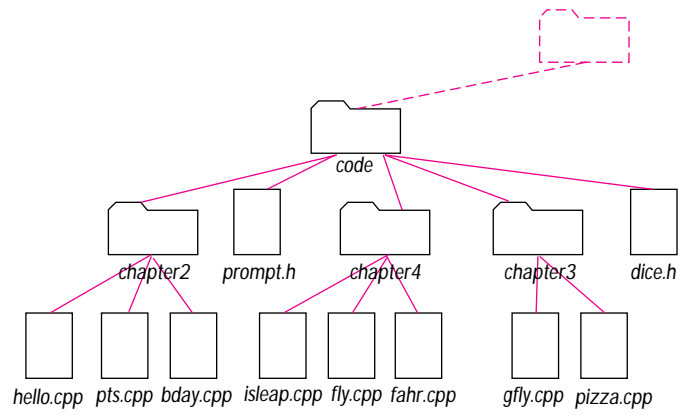


Figure 10.3:

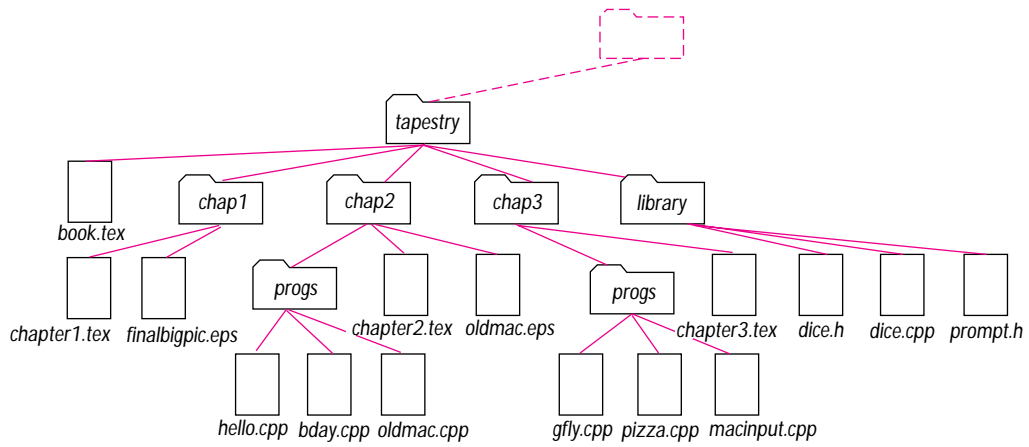


Figure 10.4:

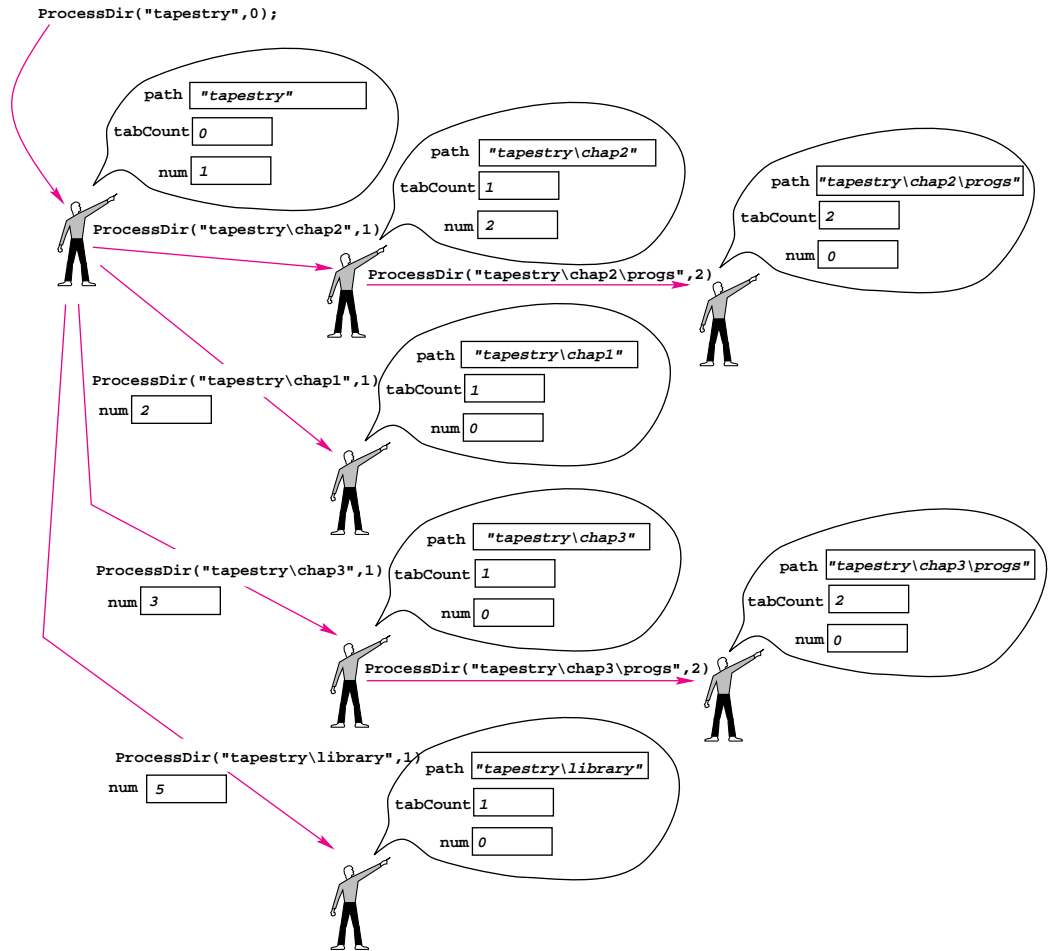


Figure 10.5:

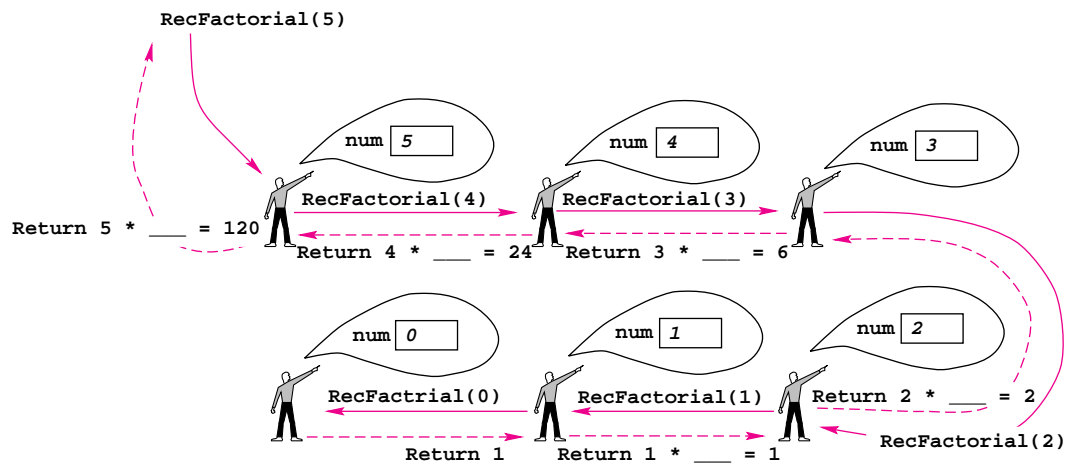


Figure 10.6:

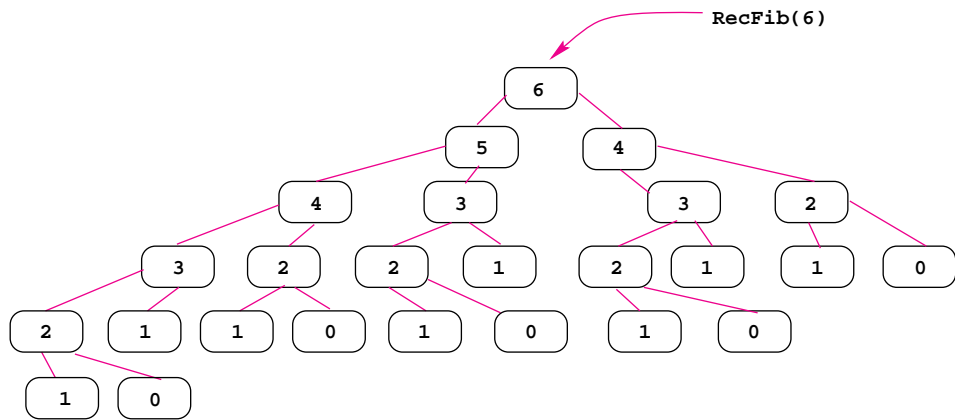


Figure 10.7:

```
main
  int first,second;
  while
    int second;
    if
      int first;
```

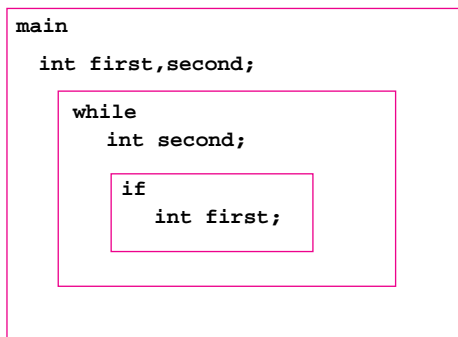
The diagram illustrates the nesting of code blocks. It consists of three nested rectangles. The outermost rectangle represents the 'main' block, containing the code 'main' and 'int first,second;'. Inside this is a 'while' block containing 'while' and 'int second;'. The innermost rectangle represents an 'if' block containing 'if' and 'int first;'. The nesting shows that the 'if' block is contained within the 'while' block, which is contained within the 'main' block.

Figure 10.8:

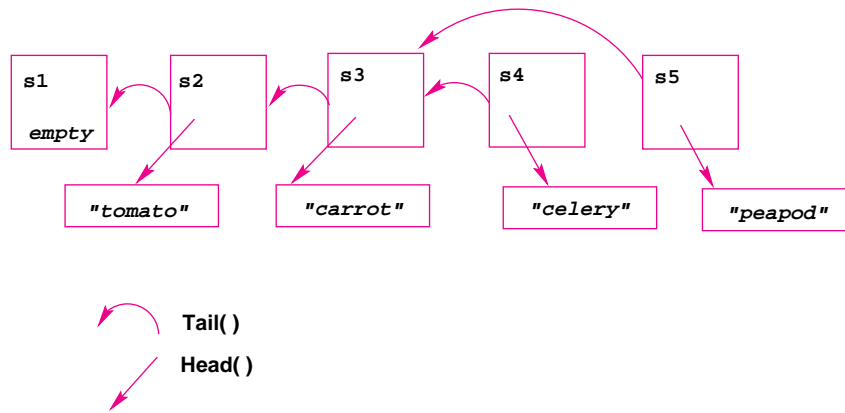


Figure 10.9:

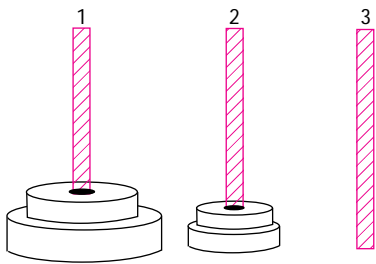


Figure 10.10:

3-neighborhood

10	12	28
10	28	25
25	32	32

5-neighborhood

10	12	12	10	10
10	10	12	32	32
25	10	28	18	18
25	25	32	32	18
32	32	32	25	25

Figure 10.12:



Figure 10.13:



Figure 10.14:

Chapter 11

Sorting, Templates, and Generic Programming

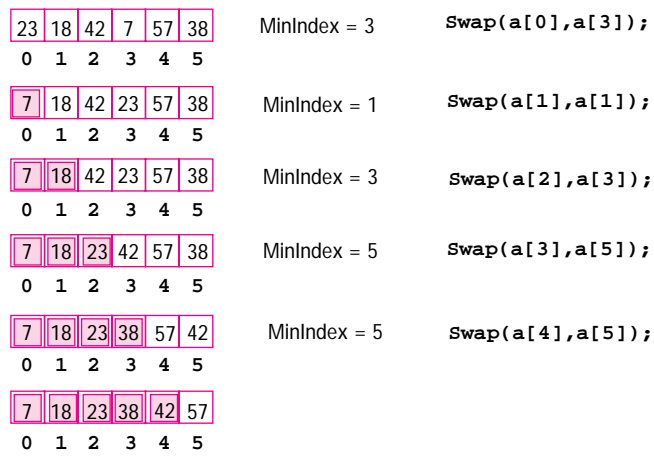


Figure 11.1:

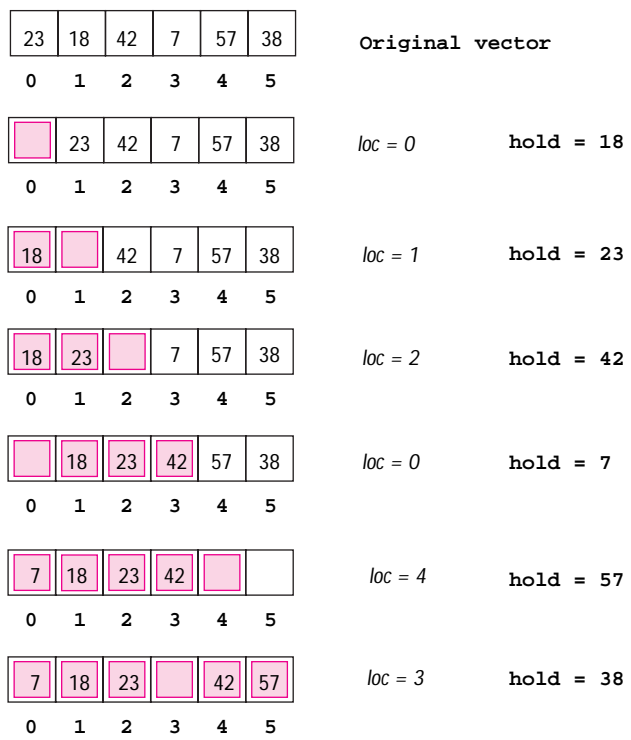


Figure 11.2:

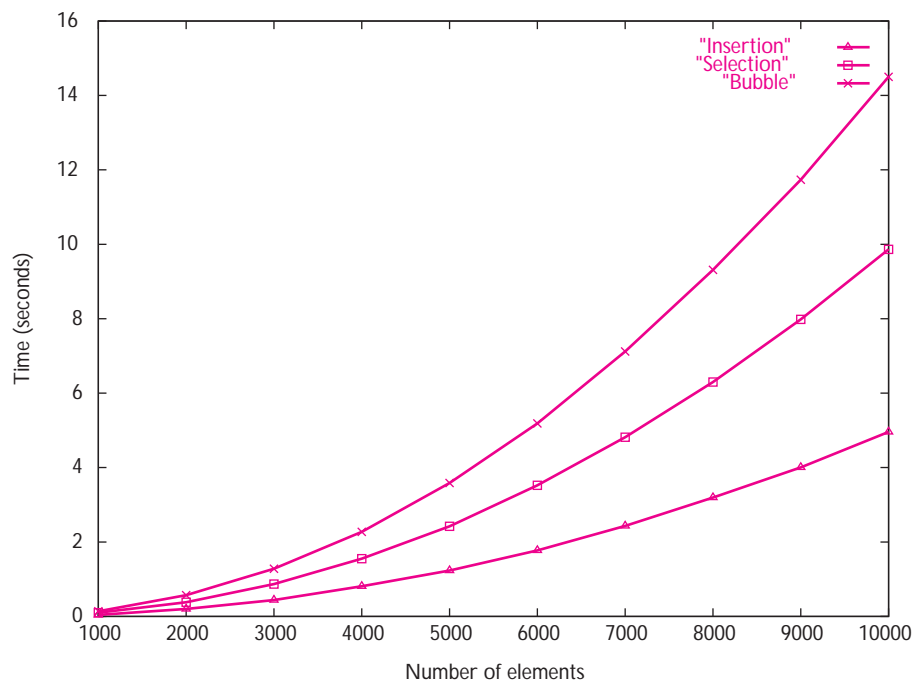


Figure 11.3:

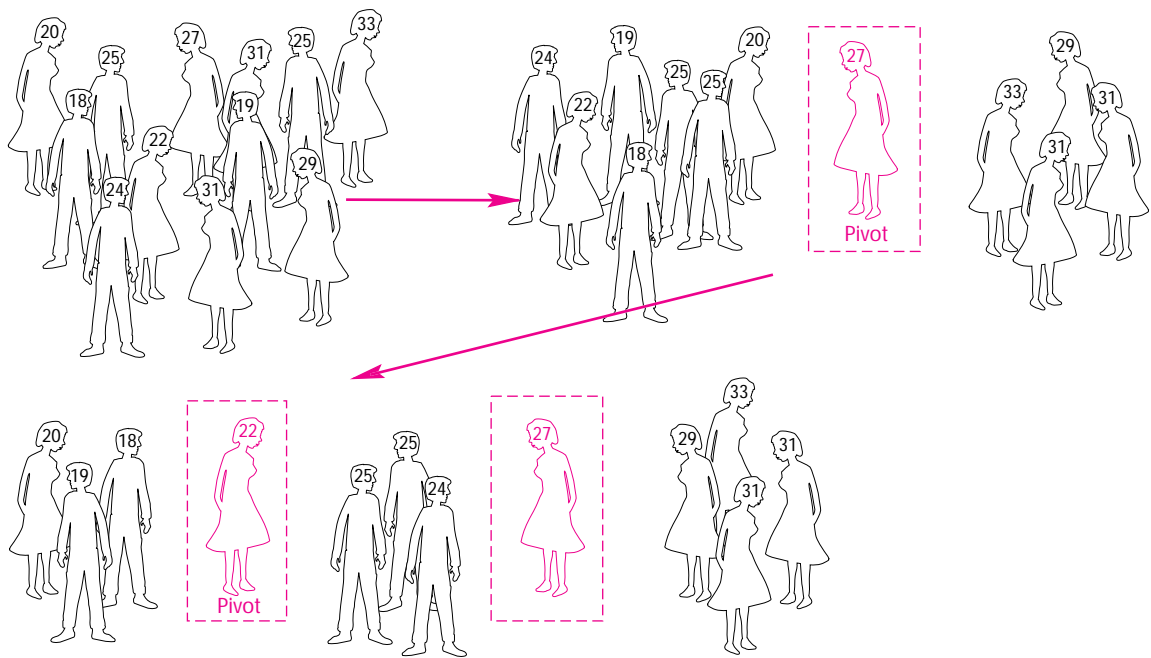
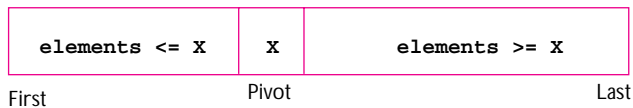
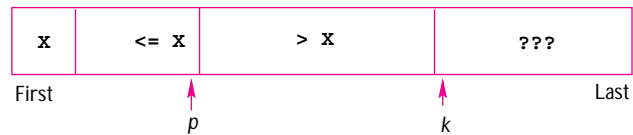


Figure 11.4:

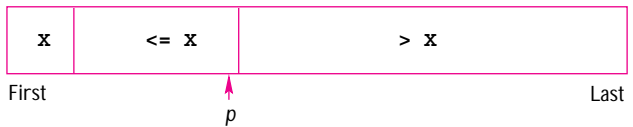
Desired properties of vector, partitioned around pivot



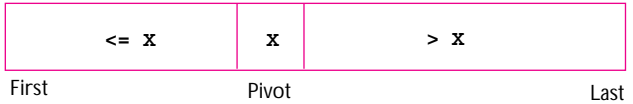
After several iterations, partially re-arranged, more elements to process



All elements processed



Final configuration after swapping first element into pivot location



```

}
if (a[k] <= piv)
{
    p++;
    Swap(a[k], a[p])
}

```

```
Swap(a[p], a[first]);
```

```
return p;
```

Figure 11.5:

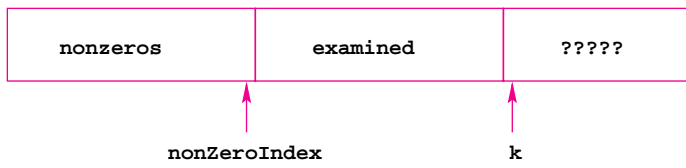


Figure 11.6: