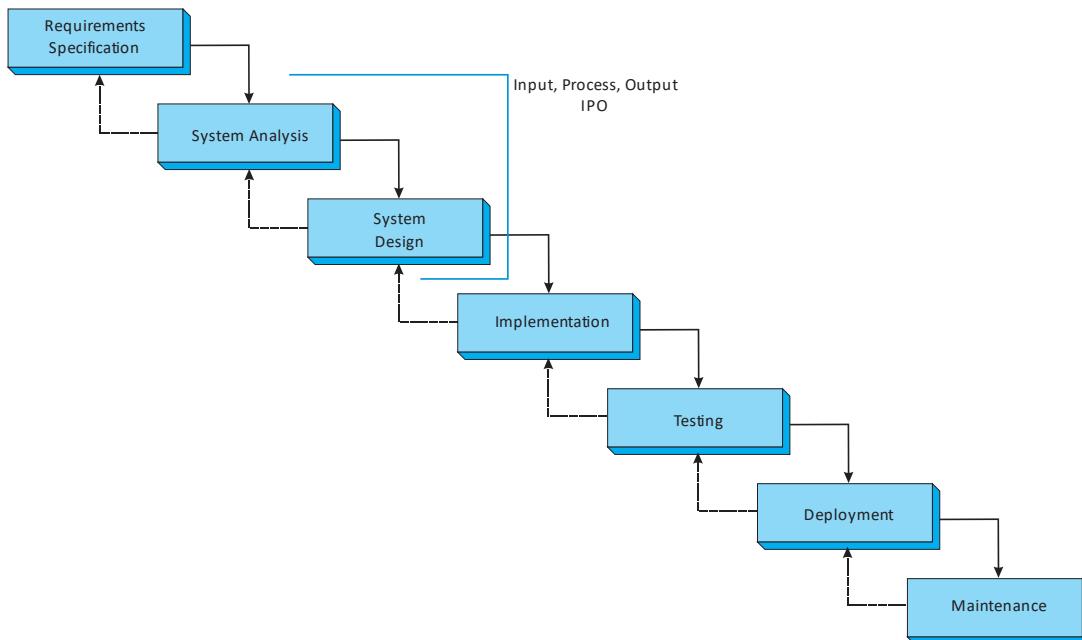


Key point: *The software development life cycle is a multistage process that includes requirements specification, analysis, design, implementation, testing, deployment, and maintenance.*



**Figure 2.3** At any stage of the software development life cycle, it may be necessary to go back to a previous stage to correct errors or deal with other issues that might prevent the software from functioning as expected.

*Requirements specification* is formal process that seeks to understand the problem the software will address, and to document in detail what the software system needs to do. This phase involves close interaction between users and developers.

*System analysis* seeks to analyze the data flow and to identify the system's input and output.

*System design* is to design a process for obtaining the output from the input. This phase involves the use of many levels of abstraction to break down the problem into manageable components and design strategies for implementing each component.

*Implementation* involves translating the system design into programs. Separate programs are written for each component then integrated to work together. This phase requires the use of a programming language such as Java. The implementation involves coding, self-testing and debugging (that is, finding errors, called *bugs*, in the code).

*Testing* ensures the code meets the requirements specification and weeds out bugs.

*Deployment* makes the software available for use. Depending on the type of software, it may be installed on each user's machine, or installed on a server accessible on the Internet.

*Maintenance* is concerned with updating and improving the product. A software product must continue to perform and improve in an ever-evolving environment.

### Stage 1: Requirements Specification

The program must satisfy the following requirements:

- It must let the user enter the interest rate, the loan amount, and the number of years for which payments will be made.
- It must compute and display the monthly payment and total payment amounts.

### Stage 2: System Analysis

The output is the monthly payment and total payment, which can be obtained using the following formula:

$$R = \frac{P_i}{1 - (1 + i)^{-n}}$$

Term	Description
$R$	(Rent) monthlyPayment
$P$	(Principal) loanAmount
$i$	(Periodic Interest Rate) monthlyInterestRate
$n$	(Term) number of months in the term. $n = 12 \times \text{numberOfYears}$

### Stage 3: System Design

During system design, you identify the steps in the program.

Step 3.1. Prompt the user to enter the annual interest rate, the number of years, and the loan amount.

(The interest rate is commonly expressed as a percentage of the principal for a period of one year. This is known as the *annual percentage rate*.)

Step 3.2. The input for the annual percentage rate is a number in percent format, such as 4.5%. The program needs to convert it to a decimal by dividing it by 100. To obtain the monthly interest rate from the annual interest rate, divide it by 12, since a year has 12 months. Thus, to obtain the monthly interest rate in decimal format, you need to divide the annual interest rate in percentage by 1200. For example, if the annual interest rate is 4.5%, then the monthly interest rate is  $\frac{4.5}{1200} = 0.00375$ .

Step 3.3. Compute the monthly payment using the preceding formula.

Step 3.4. Compute the total payment, which is the monthly payment multiplied by 12 and multiplied by the number of years.

Step 3.5. Display the monthly payment and total payment.

#### Stage 4: Implementation

Implementation is also known as *coding* (writing the code).

##### **Listing 2.9 ComputeLoan.java**

```
1 import java.util.Scanner;
2
3 public class ComputeLoan {
4     public static void main(String[] args) {
5         //Create a Scanner
6         Scanner input = new Scanner(System.in);
7
8         //Enter annual interest rate percentage. e.g., 7.25
9         System.out.print("Enter annual percentage rate. e.g., 7.25: ");
10        double annualInterestRate = input.nextDouble();
11
12        //Obtain monthly interest rate
13        double monthlyInterestRate = annualInterestRate / 1200;
14
15        //Enter number of years
16        System.out.print("Enter number of years as an integer, e.g. 5: ");
17        int numberOfYears = input.nextInt();
18
19        //Enter loan amount
20        System.out.print("Enter loan amount, e.g. 120000.95: ");
21        double loanAmount = input.nextDouble();
22
23        //Calculate payment
24        double monthlyPayment = loanAmount * monthlyInterestRate / (1
25            - 1 / Math.pow(1 + monthlyInterestRate, numberOfYears * 12));
26        double totalPayment = monthlyPayment * numberOfYears * 12;
27
28        //Display results
29        System.out.println("The monthly payment is $" +
30            (int)(monthlyPayment * 100) / 100.0);
31        System.out.println("The total payment is $" +
32            (int)(totalPayment * 100) / 100.0);
33    }
34 }
```

```
run:
Enter annual interest rate. e.g., 7.25: 5.75
Enter number of years as an integer, e.g. 5: 15
Enter loan amount, e.g. 120000.95: 250000
The monthly payment is $2076.02
The total payment is $373684.53
BUILD SUCCESSFUL (total time: 48 seconds)
```

**Stage 5: Testing**

After the program is implemented, test it with some sample input data and verify whether the output is correct. Some of the problems may involve many cases, as you will see in later chapters. For these types of problems, you need to design test data that cover all cases.