1.11 Practice Set

- Write a program to simulate a simple calculator that performs basic arithmetic operations (addition, subtraction, multiplication, and division).
- Design a controller based on the von Neumann model.
- Discuss the concept of a computer as a memory device.
- Explain the role of a computer in a control system.
- What is the role of a program in a computer?
- Which is the function of the control unit?
- What is the function of the ALU?
- Which is the function of the memory subsystem?
- Why are the various subsystems of a computer based on the von Neumann model?

1.10 Summary

The idea of a universal computational device was first proposed by Alan Turing in 1937. He proposed that all machines can be modeled by a specific type of machine, now called a Turing machine.

Turing model 1
- instruction
- control unit
- arithmetic logic unit (ALU)
- data processor
- control unit
- memory
- operating system
- input device
- output device

1.9 Key Terms

- Introduction

- Software engineering
- Hardware engineering
- control unit
- arithmetic logic unit
- data processor
- control unit
- memory
- operating system
- input device
- output device
Multiple-choice questions

11. The _____ model is the basis for today's computers.
   a. Leibnitz
   b. von Neumann
   c. Pascal
   d. Charles Babbage

12. In a computer, the _____ subsystem stores data and programs.
    a. ALU
    b. input/output
    c. memory
    d. control unit

13. In a computer, the _____ subsystem performs calculations and logical operations.
    a. ALU
    b. input/output
    c. memory
    d. control unit

14. In a computer, the _____ subsystem accepts data and programs and sends processing results to output devices.
    a. ALU
    b. input/output
    c. memory
    d. control unit

15. In a computer, the _____ subsystem serves as a manager of the other subsystems.
    a. ALU
    b. input/output
    c. memory
    d. control unit

16. According to the von Neumann model, _____ stored in memory:
    a. only data is
    b. only programs are
    c. data and programs are
    d. (none of the above)

17. A step-by-step solution to a problem is called _____.
    a. hardware
    b. an operating system
    c. a computer language
    d. an algorithm

18. FORTRAN and COBOL are examples of _____.
    a. hardware
    b. operating systems
    c. computer languages
    d. algorithms

19. A 17th-century computing machine that could perform addition and subtraction was the _____.
    a. Pascalan
    b. Jacquard loom
    c. Analytical Engine
    d. Babbage machine

20. _____ is a set of instructions in a computer language that tells the computer what to do with data.
    a. An operating system
    b. An algorithm
    c. A data processor
    d. A program

21. _____ is the design and writing of a program in structured form.
    a. Software engineering
    b. Hardware engineering
    c. Algorithm development
    d. Instructional architecture

22. The first electronic special-purpose computer was called _____.
    a. Pascal
    b. Pascalan
    c. ABC
    d. ENIAC

23. One of the first computers based on the von Neumann model was called _____.
    a. Pascal
    b. Pascalan
    c. ABC
    d. EDVAC

24. The first computing machine to use the idea of storage and programming was called _____.
    a. the Madeline
    b. EDVAC
    c. the Babbage machine
    d. the Jacquard loom
the transport layers of the client and the server computer. During the life of the TCP/IP protocol suite, three transport layer protocols have been designed: UDP, TCP, and SCTP.

- The network layer is responsible for the source-to-destination delivery of a packet, possibly across multiple networks. The network layer ensures that each packet gets from its point of origin to its final destination. In the TCP/IP protocol suite, the main protocol at the network layer is Internet Protocol (IP).

- The data-link layer delivers a packet from one node to another. The data-link layer is also responsible for error and flow control between "hops". It uses physical or MAC address to identify the nodes.

- The physical layer coordinates the functions required to carry a bit stream over a physical medium. Although the data-link layer is responsible for moving a frame from one node to another, the physical layer is responsible for moving the individual bits that enable the frame to reach the next node.

- Electronic mail (e-mail) is the most popular application on the Internet. The main protocol used for e-mail is called Simple Mail Transfer Protocol (SMTP). Other protocols used for electronic mail are POP and IMAP.

- File Transfer Protocol (FTP) is the standard mechanism for one of the most common tasks on the Internet, copying a file from one computer to another. FTP differs from other client-server applications in that it establishes two connections: one for data transfer and one for exchanging control commands.

- TELNET is a general-purpose client-server program that lets a user access any application program on a remote computer. In other words, it allows the user to log in to a remote computer. After login, a user can use the services available on the remote computer and transfer the results back to the local computer.

- The World Wide Web (WWW), or web, is a repository of information spread all over the world and linked together. To use the WWW, we need three components: a browser, a web server, and a protocol called Hypertext Transfer Protocol (HTTP).

- Documents on the WWW can be grouped into three broad categories: static, dynamic, and active. The category is based on the time at which the contents of the document are determined. A dynamic document is created by a web server whenever a browser requests the document. An active document is a program that is run at the client site.

- Three other applications were mentioned in this chapter. Videoconferencing can provide communication between two or more groups of participants or a set of individual participants. Listservs allow a group of users to discuss a common topic of interest by using two programs running on the server: the subscriber server and the mailer server. Finally, chat is supported by a class of real-time applications similar to videoconferencing, in which two or more parties are involved in an exchange of text, audio, and video.

### 6.8 PRACTICE SET

**Review questions**

1. What is the difference between a point-to-point and a multipoint connection?
2. List common physical network topologies.
3. Define the three types of network.
4. Distinguish between an internet and the Internet.
5. Name the layers of the TCP/IP protocol suite.
6. What is the main function of the application layer in the TCP/IP protocol suite? What type of addresses are used in this layer?
7. What is the main function of the transport layer in the TCP/IP protocol suite? What type of addresses are used in this layer?
8. What is the main function of the network layer in the TCP/IP protocol suite? What type of addresses are used in this layer?
Multiple-choice questions

10. The TCP/IP model has ___ levels.
   a. five
   b. six
   c. seven
   d. eight

11. Define node-to-node delivery. In which layer does this type of delivery take place?
   a. transport
   b. network
   c. data-link
   d. session

12. What is the domain name in the entire message?
   a. transport
   b. network
   c. data-link
   d. session

13. What is the purpose of FTP?
   a. transport
   b. network
   c. data-link
   d. session

14. Which physical topology uses a hub?
   a. bus
   b. ring
   c. star
   d. none of the above

15. Which physical topology uses a star?
   a. bus
   b. ring
   c. star
   d. none of the above

16. The TCP/IP protocol suite is one of the major protocols on the Internet. Which of the following is a protocol for the transfer of files?
   a. FTP
   b. SMTP
   c. HTTP
   d. All of the above

17. The transport layer provides services for end users. Which of the following is a protocol for e-mail services?
   a. FTP
   b. SMTP
   c. HTTP
   d. All of the above

18. Compare and contrast the three Internet document retrieval protocols. Which of the following is a protocol for accessing databases?
   a. FTP
   b. SMTP
   c. HTTP
   d. All of the above

19. The TCP/IP protocol suite transmits a bit stream over a physical medium. Which of the following is a protocol for accessing databases?
   a. FTP
   b. SMTP
   c. HTTP
   d. All of the above

20. What is the purpose of IPv4 addresses in the TCP/IP protocol suite?
   a. five
   b. six
   c. seven
   d. eight

21. The transport layer provides services for end users. Which of the following is a protocol for e-mail services?
   a. FTP
   b. SMTP
   c. HTTP
   d. All of the above

22. The transport layer provides services for end users. Which of the following is a protocol for accessing databases?
   a. FTP
   b. SMTP
   c. HTTP
   d. All of the above

23. The layer responsible for source-to-destination in the TCP/IP protocol suite is the: a. transport b. network c. data-link d. session

24. What is the domain name in the entire message?
   a. transport
   b. network
   c. data-link
   d. session

25. What is the purpose of FTP?
   a. transport
   b. network
   c. data-link
   d. session

26. Which physical topology uses a hub?
   a. bus
   b. ring
   c. star
   d. none of the above

27. Which physical topology uses a star?
   a. bus
   b. ring
   c. star
   d. none of the above

28. The transport layer provides services for end users. Which of the following is a protocol for the transfer of files?
   a. FTP
   b. SMTP
   c. HTTP
   d. All of the above

29. The transport layer provides services for end users. Which of the following is a protocol for e-mail services?
   a. FTP
   b. SMTP
   c. HTTP
   d. All of the above

30. The transport layer provides services for end users. Which of the following is a protocol for accessing databases?
   a. FTP
   b. SMTP
   c. HTTP
   d. All of the above
9.7 SUMMARY

- A computer language is a set of predefined words that are combined into a program according to predefined rules, the language’s syntax. Over the years, computer languages have evolved from machine language to high-level languages. The only language understood by a computer is machine language.
- High-level languages are portable to many different computers, allowing the programmer to concentrate on the application rather than the intricacies of the computer’s organization.
- To run a program on a computer, the program needs to be translated into the computer’s native machine language. The program in the high-level language is called the source program. The translated program in machine language is called the object program. Two methods are used for translation: compilation and interpretation. A compiler translates the whole source program into the object program. Interpretation refers to the process of translating each line of the source program into the corresponding object program line by line and executing them.
- The translation process uses a lexical analyzer, a syntax analyzer, a semantic analyzer, and a code generator to create a list of tokens.
- A paradigm describes a way in which a computer language can be used to approach a problem to be solved. We divide computer languages into four paradigms: procedural, object-oriented, functional, and declarative.
- The procedural paradigm considers a program as an active agent that manipulates passive objects. FORTRAN, COBOL, Pascal, C, and Ada are examples of procedural languages.
- The object-oriented paradigm deals with active objects instead of passive objects. C++ and Java are common object-oriented languages.
- In the functional paradigm, a program is considered as a mathematical function. In this context, a function is a black box that maps a list of inputs to a list of outputs. LISP and Scheme are common functional languages.
- A declarative paradigm uses the principle of logical reasoning to answer queries. One of the best-known declarative languages is PROLOG.
- Some common concepts in procedural and object-oriented languages are identifiers, data types, variables, literals, constants, inputs and outputs, expressions, and statements. Most languages use two categories of control statements: decision and repetition. Subprogramming is a common concept among procedural languages.

9.8 PRACTICE SET

Review questions

1. Distinguish between machine language and assembly language.
2. Distinguish between assembly language and a high-level language.
3. Which computer language is directly related to and understood by a computer?
4. Distinguish between compilation and interpretation.
5. List four steps in programming language translation.
7. Compare and contrast a procedural paradigm with an object-oriented paradigm.
8. Define a class and a method in an object-oriented language. What is the relation between these two concepts and the concept of an object?
10. Define a declarative paradigm.
C = 5

```c
while (a > 0)
    a = a + 2;
    statement;
}
```

Exercises:

27. Write a program that outputs the number of times the loop statements are executed. How many times does this happen in the program?

28. Explain why a constant cannot be updated in C. Use a C expression to demonstrate this.

29. Determine three constants in C that are treated as type integer in C and determine three variables of type real in C.

30. Find the answer to the following question:

```
for (int i = 0; i < 10; i++)
    printf("Hello, world!");
```

31. Identify the following code fragment in C:

```
while (a > 0)
    a = a + 2;
    statement;
}
```

32. What is the output of the following program?

```
#include <stdio.h>

int main()
{
    int a = 3;
    int b = 2;
    int c = a + b;
    printf("%d\n", c);
    return 0;
}
```

33. How can the following program be classified?

```
#include <stdio.h>

int main()
{
    int a = 3;
    int b = 2;
    int c = a + b;
    printf("%d\n", c);
    return 0;
}
```

34. What is a programming language?

```
#include <stdio.h>

int main()
{
    int a = 3;
    int b = 2;
    int c = a + b;
    printf("%d\n", c);
    return 0;
}
```

35. What is a type in C?

```
#include <stdio.h>

int main()
{
    int a = 3;
    int b = 2;
    int c = a + b;
    printf("%d\n", c);
    return 0;
}
```

36. What is a procedure in C?

```
#include <stdio.h>

int main()
{
    int a = 3;
    int b = 2;
    int c = a + b;
    printf("%d\n", c);
    return 0;
}
```

37. What is a function in C?

```
#include <stdio.h>

int main()
{
    int a = 3;
    int b = 2;
    int c = a + b;
    printf("%d\n", c);
    return 0;
}
```

38. What is a program in C?

```
#include <stdio.h>

int main()
{
    int a = 3;
    int b = 2;
    int c = a + b;
    printf("%d\n", c);
    return 0;
}
```
10.9 SUMMARY

The software lifecycle is a fundamental concept in software engineering. It is divided into phases that go through a cycle of repeating steps. These phases include analysis, design, implementation, and testing. Several models have been described in relation to these phases. We discussed the two most common: the waterfall model and the incremental model.

10.10 PRACTICE SET

1. Define "software lifecycle."  
2. Distinguish between the waterfall model and the incremental development model.
3. List the four phases in the development process.
4. Define the purpose of the analysis phase.
5. Describe two trends in the design phase.
6. Describe two trends in the implementation phase.
7. Distinguish between coupling and cohesion.
8. Define the purpose of the implementation phase.
9. Define the purpose of the testing phase (a) testing all categories of testing, (b) testing black box testing, (c) testing white box testing.
10. Distinguish between black box testing and white box testing.

Review questions:

1. Define the purpose of the testing phase.
2. Describe two trends in the design phase.
3. Describe two trends in the implementation phase.
4. Define the purpose of the analysis phase.
5. Describe two trends in the design phase.
6. Describe two trends in the implementation phase.
7. Distinguish between coupling and cohesion.
8. Define the purpose of the implementation phase.
9. Define the purpose of the testing phase.
10. Distinguish between black box testing and white box testing.

Multiple-choice questions:

(a) All of the above  
(b) definition phase  
(c) analysis  
(d) design  
(e) implementation  
(f) testing  
(g) design  
(h) analysis  
(i) implementation  
(j) testing  
(k) definition phase

9. Define the purpose of the testing phase.
10. Distinguish between black box testing and white box testing.

Analysis:

1. Define the purpose of the analysis phase.
2. Describe two trends in the design phase.
3. Describe two trends in the implementation phase.
4. Define the purpose of the analysis phase.
5. Describe two trends in the design phase.
6. Describe two trends in the implementation phase.
7. Distinguish between coupling and cohesion.
8. Define the purpose of the implementation phase.
9. Define the purpose of the testing phase.
10. Distinguish between black box testing and white box testing.

Design:

1. Define the purpose of the design phase.
2. Describe two trends in the implementation phase.
3. Describe two trends in the testing phase.
4. Define the purpose of the design phase.
5. Describe two trends in the implementation phase.
6. Describe two trends in the testing phase.
7. Distinguish between coupling and cohesion.
8. Define the purpose of the implementation phase.
9. Define the purpose of the testing phase.
10. Distinguish between black box testing and white box testing.

Implementation:

1. Define the purpose of the implementation phase.
2. Describe two trends in the testing phase.
3. Describe two trends in the design phase.
4. Define the purpose of the implementation phase.
5. Describe two trends in the testing phase.
6. Describe two trends in the design phase.
7. Distinguish between coupling and cohesion.
8. Define the purpose of the implementation phase.
9. Define the purpose of the testing phase.
10. Distinguish between black box testing and white box testing.

Testing:

1. Define the purpose of the testing phase.
2. Describe two trends in the design phase.
3. Describe two trends in the implementation phase.
4. Define the purpose of the testing phase.
5. Describe two trends in the design phase.
6. Describe two trends in the implementation phase.
7. Distinguish between coupling and cohesion.
8. Define the purpose of the implementation phase.
9. Define the purpose of the testing phase.
10. Distinguish between black box testing and white box testing.
13. In the system development process, writing the program is part of the ______ phase.
   a. analysis
   b. design
   c. implementation
   d. testing

14. In the system development process, structure charts are tools used in the ______ phase.
   a. analysis
   b. design
   c. implementation
   d. testing

15. Testing a software system can involve ______
   a. black-box
   b. glass-box
   c. neither a nor b
   d. a and b

16. ______ is the breaking up of a large project into smaller parts.
   a. Coupling
   b. Incrementing
   c. Obsolescence
   d. Modularization

17. ______ is a measure of how tightly two modules are bound to each other.
   a. Modularity
   b. Coupling
   c. Interoperability
   d. Cohesion

18. ______ is a measure of how closely the processes in a program are related.
   a. Modularity
   b. Coupling
   c. Interoperability
   d. none of the above

19. ______ between modules in a software system must be minimized.
   a. Coupling
   b. Cohesion
   c. neither a nor b
   d. a and b

20. ______ between modules in a software system must be maximized.
   a. Coupling
   b. Cohesion
   c. neither a nor b
   d. a and b

Exercises
21. In Chapter 9 we explained that the use of constant values are preferred to literals. What it the effect of this preference on the software lifecycle?
22. In Chapter 9 we showed that communication between two modules can take place either by pass-by-value or pass-by-reference. Which method provides less coupling between the two modules?
23. In Chapter 9 we showed that communication between two modules can take place either by pass-by-value or pass-by-reference. Which method provides more cohesion between the two modules?
24. Draw a use case diagram for a simple library.
25. Draw a use case diagram for a small grocery store.
26. Show the data flow diagram for a simple mathematical formula x + y.
27. Show the data flow diagram for a simple mathematical formula x * y + z * t.
28. Show the data flow diagram for a library.
29. Show the data flow diagram for a small grocery store.
30. Create a structure chart for Exercise 28.
31. Create a structure chart for Exercise 29.
32. Show a state diagram for a stack of fixed capacity (see Chapter 12).
33. Show a state diagram for a queue of fixed capacity (see Chapter 12).
34. Create a class diagram for a library.
35. Create a class diagram for a small grocery store.
36. Show the details of classes in Exercise 34.
37. Show the details of classes in Exercise 35.
38. The input data to a program is made up of a combination of three integers in the range 1000 to 1999 (inclusive). Find the number of exhaustive tests to test all combinations of these numbers.