

## Chapter 1 Answers

### Exam Preparation Exercises

1. a: v, b: i, c: viii, d: iii, e: iv, f: vii, g: vi, h: ii.
2. Analysis and specification, general solution (algorithm), verify.
3. Concrete solution (program), test.
4. The analysis and specification step within the problem-solving phase.
5. The steps never end. Changing the last step to say, "Repeat from first step until graduation," converts the sequence into an algorithm that eventually ends, assuming the person will graduate from school some day. We can make sure that this loop ends by specifying a condition that will definitely occur. For example, "Repeat from first step until last day of classes."
6. a: v, b: ii, c: iv. d: i, e: vi, f: iii.
7. The program can be compiled on different computer systems without modification.
8. The control unit directs the actions of the other components in the computer to execute program instruction in the proper order.
9. False. The editor is software rather than hardware.
10. False. Peripheral devices are external to the CPU and its main memory.
11. Yes. If the software license restricts use to a single computer or a single user, then this is a case of piracy, even though you split the cost.
12. a: ii, b: v, c: iii, d: vii, e: i, f: vi, g: iv.

### Programming Warm-Up Exercises

1. Branches: Steps n1, n2, and n3. There are two loops: steps p, q, and r; and steps t and u. Steps a, c, and e are references to subalgorithms defined elsewhere.

2.

Follow separate algorithm for filling glass with water and placing on counter

If right handed,

pick up toothpaste tube in left hand and unscrew cap by turning counter-clockwise

place toothpaste cap on counter

transfer toothpaste tube to right hand and pick up toothbrush in left hand, holding it by its handle

# Contact me in order to access whole complete document

place open end of toothpaste tube against tips of brush bristles and squeeze toothpaste tube just enough to create a 0.5 inch long extrusion of toothpaste on the brush.

pull toothpaste tube away from brush and place tube on counter

transfer toothbrush to right hand, holding it by its handle

open mouth and insert brush, placing toothpaste-covered bristles against one tooth

scrub tooth with brush by moving brush up and down ten times

reposition brush to an unscrubbed tooth

repeat previous two steps until all teeth have been brushed

spit toothpaste into sink

put toothbrush in sink

pick up glass in right hand

fill mouth with water from glass

swish water in mouth for five seconds

spit water from mouth into sink

repeat previous three steps three times

Otherwise (if left-handed)

pick up toothpaste tube in right hand and unscrew cap by turning counter-clockwise

place toothpaste cap on counter

transfer toothpaste tube to left hand and pick up toothbrush in right hand, holding it by its handle

place open end of toothpaste tube against tips of brush bristles and squeeze

toothpaste tube just enough to create a 0.5 inch long extrusion of toothpaste on the brush.

pull toothpaste tube away from brush and place tube on counter

transfer toothbrush to left hand, holding it by its handle

open mouth and insert brush, placing toothpaste-covered bristles against one tooth

scrub tooth with brush by moving brush up and down ten times

reposition brush to an unscrubbed tooth

repeat previous two steps until all teeth have been brushed

spit toothpaste into sink

put toothbrush in sink

pick up glass in right hand

fill mouth with water from glass

swish water in mouth for five seconds

spit water from mouth into sink

repeat previous three steps three times

Place glass on counter

Follow separate algorithm for cleaning up after brushing teeth

3. The test for right or left handedness is a branch. In each branch there are two loops, one that iterates until all teeth are brushed, and the other iterates rinsing four times. The first and last step refer to separately defined subalgorithms.

4. Change step u to:

u. Repeat step t nine times.

## Chapter 2 Answers

### Exam Preparation Exercises

1. a. valid, b. invalid (hyphens not allowed), c. invalid (reserved words not allowed), d. valid, e. valid, f. valid, g. invalid (must begin with a letter), h. invalid (# is not allowed).
2. a: vi, b: ix, c: iv, d. v, e: vii, f: ii, g: i, h: x, i: iii, j: viii.
3. False. Reserved words are used in many ways in C++.
4. The template allows an identifier to begin with a digit. Identifiers can begin with a letter or underscore, and digits may only appear in the second character position and beyond.
5. False. Only single quotes enclose a `char` literal. Double quotes around a single character define a string literal.
6. True.
7. True.
8. Note that the second line concatenates two words without a separating space.

```
Four score and  
seven years agoour fathers  
brought forth on this  
continent a new nation...
```

9. The following would all be displayed on a single line:

```
Bjarne Stroustrup named his new programming language C++  
because it is a successor to the C programming language.
```

10. By preceding it with a backslash character (`\`).
11. If we forget the `*/` at the end of a comment, then any amount for program code can be inadvertently included in the comment. The `//` form avoids this by automatically terminating at the end of the line.
12. The `//` form of comment cannot span more than one line. It also cannot be inserted into the middle of a line of code because everything to the right of `//` becomes a comment.
13. `<<` is the stream insertion operator, and is pronounced "put to" or "is sent," as in "cout is sent string4."

14. No. The endl identifier is a manipulator, and is not a string value.

15. It tells the C++ preprocessor to insert the contents of a specified file at that point in the code.

16. Omission of the following preprocessor directive at the start of the program:

```
#include <iostream>
```

17. A block.

18. By splitting it into pieces that fit on a line, and joining them together with the concatenation operator.

19.

```
#include <iostream>
#include <string>

using namespace std;

const string TITLE = "Dr.";
int main()
{
    cout << "Hello " + TITLE + " Stroustrup!";
    return 0;
}
```

### Programming Warm-Up Exercises

1. Substitute the actual data in the following:

```
cout << "July 4, 2025" << endl;
```

2. Note that the \" escape sequence is needed in six places here:

```
cout << "He said, \"How is that possible?\"" << endl
    << "She replied, \"Using manipulators.\"" << endl
    << "\"Of course,\" he exclaimed!" << endl;
```

3. a. const string ANSWER = "True";

b. char middleInitial;

c. string courseTitle;

d. `const char PERCENT = '%'`

4.

```
const string FIRST = "Your first name inserted here";
const string LAST = "Your last name inserted here";
// Insert your middle initial in place of A:
const char MIDDLE = 'A';
```

5. `cout << PART1 << PART2 << PART1 << PART3;`

6. `PART1 + PART2 + PART1 + PART3`

7.

```
cout << "Yet the web of thought has no such creases" <<
    endl;
cout << "And is more like a weaver's masterpieces;" <<
    endl;
cout << "One step a thousand threads arise," << endl;
cout << "Hither and thither shoots each shuttle," << endl;
cout << "The threads flow on unseen and subtle," << endl;
cout << "Each blow effects a thousand ties." << endl;
```

8.

```
#include <iostream>
#include <string>
```

```
using namespace std;
```

```
const string TITLE = "Rev.";
const char FIRST = 'H';
const char MID 'G';
const char DOT = '.';
```

```
int main()
```

```
{
    cout << TITLE << FIRST << DOT << MID << DOT << " Jones";
}
```

9. The solution to this problem is for the student to show that he or she has run the program correctly. It should output:

```
*****
*                                     *
*   Welcome to C++ Programming!     *
*                                     *
*****
```

10. The missing items are underlined.

```
#include <iostream>
#include <string>
using namespace std;
const string A = "A";
const string B = "B";
const char C = 'C';
int main (()
{
    string forward;
    string backward;
    forward = A + " " + B + " " + C;
    cout << "Forward order: " << forward << endl;
    backward = C + " " + B + " ";
    cout << "Backward order: ";
    cout << backward << A << endl;
    return 0;
}
```