Chapter 1

1. a. true; b. true; c. false; d. true; e. true; f. true; g. false; h. true; i. true; j. false; k. true; l. true; m. false; n. false

2. Hardware and software.

3. All programs must be loaded into main memory before they can be executed. Similarly, all data must be brought into main memory before a program can manipulate it.

4. Secondary storage is used to store longer-term information.

5. The operating system monitors the overall activity of the computer and provides services.

6. System programs and application programs.

7. In machine language the programs are written using the binary codes while in high-level language the program are closer to the natural language. For execution, a high-level language program is translated into the machine language while a machine language need not be translated into any other language.

8. The equivalent machine language program is of a high-level language program.

9. To load an executable program into main memory.

10. In linking, an object program is combined with other programs in the library, used in the program, to create the executable code.

11. A compiler reports syntax errors.

12. A step-by-step problem-solving process in which a solution is arrived at in a finite amount of time.

13. a. Analyze the problem, outline the problem and its solution requirements, and design an algorithm to solve the problem.

b. Implement the algorithm in a programming language, such as C++, and verify that the algorithm works.

c. Maintain the program by using and modifying it if the problem domain changes.

14. A well-analyzed problem leads to a well-designed algorithm. Moreover, a program that is well analyzed is easier to modify as well as spot and fix errors.

15. To find the weighted average of the four test scores, first you need to know each test score and its weight. Next, you multiply each test score with its weight, and then add these numbers to get the average. Therefore,

1. Get testScore1, weightTestScore1
2. Get testScore2, weightTestScore2
3. Get testScore3, weightTestScore3
4. Get testScore4, weightTestScore4
5. weightedAverage = testScore1 \* weightTestScore1 +

testScore2 \* weightTestScore2 +

testScore3 \* weightTestScore3 +

testScore4 \* weightTestScore4;

16. a. Get quarters

b. Get dimes

c. Get nickels

d. Get pennies

e. changeInPennies = quarters \* 25 + dimes \* 10 + nickels \* 5

+ pennies

17. To find the price per square inch, first we need to find the area of the pizza. Then we divide the price of the pizza by the area of the pizza. Let radius denote the radius and area denote the area of the circle, and price denote the price of pizza. Also, let pricePerSquareInch denote the price per square inch.

a. Get radius

b. area = π \* radius \* radius

c. Get price

d. pricePerSquareInch = price / area

18. To calculate the selling price of an item, we need to know the original price (the price the store pays to buy) of the item. We can then the use the following formula to find the selling price:

sellingPrice = (originalPrice + originalPrice × 0.80) × 0.90

The algorithm is as follows:

1. Get originalPrice
2. Calculate the sellingPrice using the formula:

sellingPrice = (originalPrice + originalPrice × 0.80) × 0.90

The information needed to calculate the selling price is the original price and the marked-up percentage.

19. Suppose that radius denotes radius of the sphere, volume denotes volume of the sphere, and surfaceArea denotes the surface area of the sphere. The following algorithm computes the volume and surface area of the sphere.

|  |  |
| --- | --- |
| **Algorithm** | **C++ Instruction (Code)** |
| 1 1. Get the radius. | cin >> radius; |
| 2 2. Calculate the volume. | volume = (4.0 / 3.0) \* 3.1416 \* radius \* radius \* radius; |
| 3 3. Calculate the surface area. | surfaceArea = 4.0 \* 3.1416 \* radius \* radius; |

20. Suppose that billingAmount denotes the total billing amount, movingCost denotes moving cost, area denotes the area of the yard that needs to be moved, numOfAppl denotes the number of fertilizing applications, and numOfTrees denotes the number of trees to be planted. The following algorithm computes and outputs the billing amount.

1. Enter the area of the yard.
2. Get area
3. Enter the number of fertilizing applications.
4. Get numOfAppl
5. Enter the number of trees to be planted.
6. Get numOfTrees
7. Calculate the billingAmount using the formula:

billingAmount = (area / 5000) \* 35.00 + numOfAppl \* 30.00

+ numOfTrees \* 50.00

21. Suppose that billingAmount denotes the total billing amount, numOfItemsOrdered denotes the number of items ordered, shippingAndHandlingFee denotes the shipping and handling fee, and price denotes the price of an item. The following algorithm computes and outputs the billing amount.

a. Enter the number of items bought.

b. Get numOfItemsOrdered

c. billingAmount = 0.0;

d. shippingAndHandlingFee = 0.0;

e. Repeat the following for each item bought.

i. Enter the price of the item

ii. Get price

iii. billingAmount = billingAmount + price;

f. if billingAmount < 200

shippingAndHandlingFee = 10 \* numOfItemsOrdered;

g. billingAmount = billingAmount + shippingAndHandlingFee;

i. Print billingAmount

22. Suppose amountWithdrawn denotes the amount to be withdrawn, serviceCharge denotes the service charges, if any, and accountBalance denotes the total money in the account.

You can now write the algorithm as follows:

a. Get amountWithdrawn.

b. if amountWithdrawn > 500

Print "The maximum amount that can be drawn is $500"

otherwise (if accountBalance <= 0)

Print "Account balance is <= 0. You cannot withdraw any money. "

otherwise

{

if (amountWithdrawn > accountBalance)

{

Print "Insufficient balance. If you withdraw, services charges

will be $25.00. Select Yes/No."

if (Yes)

{

if (amountWithdrawn > 300)

serviceCharge = (amountWithdrawn – 300) \* 0.04;

otherwise

serviceCharge = 0;

accountBalance = accountBalance – amountWithdrawn

- serviceCharge – 25;

Print "Collect your money. "

}

}

}

otherwise

{

if (amountWithdrawn > 300)

serviceCharge = (amountWithdrawn – 300) \* 0.04;

otherwise

serviceCharge = 0;

accountBalance = accountBalance – amountWithdrawn

- serviceCharge;

Print "Collect your money."

}

23. Suppose x1 and x2 are the real root of the quadratic equation.

a. Get a

b. Get b

c. Get c

d. if (b \* b – 4 \* a \* c < 0)

Print "The equation has no real roots."

Otherwise

{

temp = b \* b – 4 \* a \* c;

x1 = (-b + temp) / (2 \* a);

x2 = (-b - temp) / (2 \* a);

}

24. Suppose that tuition denotes the tuition per semester, semesterUnits denotes the average semester units, courseUnits denotes the number of course units, semesterWeeks denotes the number of weeks in a semester, classDaysPerWeek denotes the number of days a class meets in a week, classDaysPerSemester denotes the total number of days a class meets in a semester, costPerUnit denotes the cost of one unit of a course, and costPerClass denotes the cost of one class.

a. Get tuition

b. Get semesterUnits

c. Get semesterWeeks

d. Get courseUnits

e. Get classDaysPerWeek

f. Compute costPerUnit using the following formula.

costPerUnit = tuition / semesterUnits;

g. Compute classDaysPerSemester using the following formula.

classDaysPerSemester = classDaysPerWeek \* semesterWeeks;

h. Compute costPerClass using the following formula.

costPerClass = (costPerUnit \* courseUnits) / classDaysPerSemester;

25. Suppose averageTestScore denotes the average test score, highestScore denotes the highest test score, testScore denotes a test score, sum denotes the sum of all the test scores, and count denotes the number of students in class, and studentName denotes the name of a student.

a. First you design an algorithm to find the average test score. To find the average test score, first you need to count the number of students in the class and add the test score of each student. You then divide the sum by count to find the average test score. The algorithm to find the average test score is as follows:

i. Set sum and count to 0.

ii. Repeat the following for each student in class.

1. Get testScore

2. Increment count and update the value of sum by adding the current test score to sum.

iii. Use the following formula to find the average test score.

if (count is 0)

averageTestScore = 0;

otherwise

averageTestScore = sum / count;

b. The following algorithm determines and prints the names of all the students whose test score is below the average test score.

Repeat the following for each student in class:

i. Get studentName and testScore

ii.

if (testScore is less than averageTestScore)

print studentName

c. The following algorithm determines and highest test score

i. Get first student’s test score and call it highestTestScore.

ii. Repeat the following for each of the remaining student in class

1. Get testScore

2.

if (testScore is greater than highestTestScore)

highestTestScore = testScore;

d. To print the names of all the students whose test score is the same as the highest test score, compare the test score of each student with the highest test score and if they are equal print the name. The following algorithm accomplishes this

Repeat the following for each student in class:

i. Get studentName and testScore

ii.

if (testScore is equal to highestTestScore)

print studentName

You can use the solutions of the subproblems obtained in parts a to d to design the main algorithm as follows:

1. Use the algorithm in part a to find the average test score.

2. Use the algorithm in part b to print the names of all the students whose score is below the average test score.

3. Use the algorithm in part c to find the highest test score.

4. Use the algorithm in part d to print the names of all the students whose test score is the same as the highest test score

Chapter 2

1. a. false; b. false; c. false; d. true; e. true; f. false; g. true; h. true; i. false; j. true; k. false

2. b, e, j

3. c, f

4. A keyword is a reserved word and is defined by the system. A keyword cannot be redefined in a program. A user-defined identifier can be redefined.

5. The identifiers quizNo1 and quizno1 are not the same. C++ is case sensitive. The fifth letter of quizNo1 is uppercase N while the fifth character of quizno1is lowercase n. So these identifiers are different

6 a. 31

b. 1

c. -6

d. -2

e. 9.25

f. 12

g. -42

h. 0

i. 21.75

7. a. 3

b. -0.5

c. 7.85714

d. Not possible. Both the operands of the operator % must be integers. Because the first operand, x \* z, is a floating-point value, the expression is invalid.

e. 28.5

f. Not possible. Both the operands of the operator % must be integers. Because the second operand, z, is a floating-point value, the expression is invalid.

g. Not possible. Both the operands of the operator % must be integers. Because the second operand, y + z, is a floating-point value, the expression is invalid.

h. 3.57143

8. a, b, c, e, i, j, and k are valid;

d, f, and g are invalid because the left side of an expression must be a variable. h is invalid because the operands of the mod operator must be integers.

9. x = 5, y = 7, z = 4, w = 3, t = 6

10. Variable declarations in Lines 1, 6 and 7 are correct.

Variable declaration in Line 2 is incorrect. Now, B+ is a string, so it must be enclosed in double quotes mark. Also, grade is a char variable and a string cannot be assigned to a char variable. A correct declaration is:

char grade = 'B'; //Line 2

Variable declaration in Line 3 is incorrect because the left side of the assignment operator must be a variable, and the semicolon at the end of the statement is missing. A correct declaration is:

double num = 28.5; //Line 3

The variable declaration in Line 4 is incorrect because strings are enclosed in double quotation marks. A correct declaration is:

string message = "First C++ course"; //Line 4

The variable declaration in Line 5 is incorrect because the value assigned to age must be an int value. A correct declaration is:

int age = 18; //Line 5

11. b and c are valid

12. a. int x, y;

x = 25;

y = 18;

b. int temp = 10;

char ch = 'A';

c. x = x + 5;

d. double payRate = 12.5;

e. tempNum = firstNum;

f. temp = x;

x = y;

y = temp;

g. cout << x << " " << y << " " << x + 12 / y - 18 << endl;

h. char grade = 'A';

i. int num1, num2, num3, num4;

j. x = static\_cast<int>(z + 0.5);

13. a. 5.0 / 9 \* (F – 32)

b. 'A'

c. grade = "B+";

d. str = "seventh edition";

e. totalInches = 12 \* feet + inches;

f. i = i + 1;

g. v = 4 / 3 \* (3.1416 \* r \* r \* r);

h. s = 2 \* (3.1416 \* r \* r) + 2 \* (3.1416 \* r) \* h;

i. (a + b) / c \* (e \* f) – g \* h

j. (–b + (b \* b – 4 \* a \* c)) / (2 \* a)

14. x = -6

y = 30

z = 55

w = -11

15. x = 101

y = 28

z = 28

w = 117.5

t = 29.25

16. a. x = 8, y = 3, z = 5

b. x + 3 \* y = 17

c. Product of 8 and 5 is 40

d. x + y / z = 8

e. 2 times 8 = 16

17. a. -40

b. 11.5

c. 17.75

d. 18.7

e. 20

f. 88.25

g. 4.0

18. a. cout << endl; or cout << "\n"; or cout << '\n';

b. cout << "\t";

c. cout << "\"";

19. a and c are correct

20. a. string firstName = "Susan Miller";

b. double discountedPrice = 24.50;

c. int numOfJuiceBottles = 20;

d. double milesTravelled = 135.50

e. double highestTestScore = 95.5;

21. a. int num1;

int num2;

b. cout << "Enter two numbers separated by spaces." << endl;

c. cin >> num1 >> num2;

d. cout << "num1 = " << num1 << "num2 = " << num2

<< "2 \* num1 – num2 = " << 2 \* num1 – num2 << endl;

22. A correct answer is:

#include <iostream>

#include <string>

using namespace std;

const double DECIMAL = 5.50;

const string blanks = " ";

double PAY\_RATE = 10.75;

int main()

{

int height, weight;

double discount;

double billingAmount;

double bonus;

int hoursWorked = 45;

double price;

height = 6;

weight = 156;

cout << height << " " << weight << endl;

discount = (2 \* height + weight) % 10;

price = 49.99;

billingAmount = price \* (1 - discount) - DECIMAL ;

// DECIMAL = 7.55;

cout << price << blanks << "$" << billingAmount << endl;

bonus = hoursWorked \* PAY\_RATE / 50;

cout << "Bonus = " << bonus << endl;

return 0;

}

23. A correct answer is:

#include <iostream>

using namespace std;

const char STAR = '\*';

const int PRIME = 71;

int main()

{

int count, sum;

double x;

int newNum; //declare newNum

count = 1;

sum = count + PRIME;

x = 25.67; // x = 25.67;

newNum = count \* 1 + 2; //newNum = count \* ONE + 2;

sum++; //(x + sum)++;

sum = sum + count; //sum + count = sum;

x = x + sum \* count; // x = x + sum \* COUNT;

sum += 3; //sum += 3--;

cout << " count = " << count << ", sum = " << sum

<< ", PRIME = " << PRIME << endl;

return 0;

}

24. A correct answer is:

#include <iostream>

#include <string>

using namespace std;

int main()

{

int num1, num2;

string str1;

cout << "Enter a string without any blanks in it : ";

cin >> str1;

cout << endl;

cout << "Enter two integers: ";

cin >> num1 >> num2;

cout << endl;

cout << str1 << " "

<< "num1 \* num2 = " << num1 \* num2 << endl;

return 0;

}

25. A variable must be declared before it can be used.

26. b.

27. a. x \*= 2;

b. x += y - 2;

c. sum += num;

d. z \*= x + 2;

e. y /= x + 5;

28. a. x = x + 5 – z;

b. y = y \* (2 \* x + 5 – z);

c. w = w + 2 \* z + 4;

d. x = x – (z + y – t);

e. sum = sum + num;

29.

a b c

a = (b++) + 3; 9 7 und

c = 2 \* a + (++b); 9 8 26

b = 2 \* (++c) – (a++); 10 45 27

30.

a b c sum

sum = a + b + c; 3 5 14.1 22

c /= a; 3 5 4.7 22

b += c – a; 3 6 4.7 22

a \*= 2 \* b + c; 50 6 4.7 22

31. (The user input is shaded.)

firstNum = 62

Enter three numbers: 35 10.5 27

The numbers you entered are 35, 10.5, and 27

z = 33

Enter grade: B

The letter that follows your grade is: C

32. (The user input is shaded.)

Enter last name: Miller

Enter a two digit integer: 34

Enter a decimal number: 62.5

Name: Miller

Id: 34

Mystery number: -5.14286

33.

#include <iostream>

#include <string>

using namespace std;

const double X = 13.45;

const int Y = 34;

const char BLANK = ' ';

int main()

{

string firstName, lastName;

int num;