**Chapter 1 Checkpoints:**

**1.1 Why is the computer used by so many different people, in so many different**

**professions?**

A: Because the computer can be programmed to do so many different tasks.

**1.2 List the the major hardware components of a computer system.**

A: The Central Processing Unit (CPU), main memory, secondary storage devices, input

devices, output devices.

**1.3 Internally, the CPU consists of what two units?**

A: Arithmetic and Logic Unit (ALU), and Control Unit

**1.4 Describe the steps in the fetch/decode/execute cycle**.

A:

Fetch: The CPU’s control unit fetches the program’s next instruction from main

memory.

Decode: The control unit decodes the instruction, which is encoded in the form of a

number. An electrical signal is generated.

Execute: The signal is routed to the appropriate component of the computer, which

causes a device to perform an operation.

**1.5 What is a memory address? What is its purpose**?

A: A unique number assigned to each section of memory.

**1.6 Explain why computers have both main memory and secondary storage.**

A: Program instructions and data are stored in main memory while the program is operating.

Main memory is volatile, and loses its contents when power is removed from the

computer. Secondary storage holds data for long periods of time—even when there is no

power to the computer.

**1.7 What are the two general categories of software?**

A: Operating Systems and Application Software

**1.8 What fundamental set of programs control the internal operations of the**

**computer s hardware?**

A: The operating system

**1.9 What do you call a program that performs a specialized task, such as a virus program**

scanner, a le-compression program, or a data-backup program?

A: A utility program

**1.10 Word processing programs, spreadsheet programs, e-mail programs, Web browsers,**

**and game programs belong to what category of software?**

A: application software

**1.11 What is an algorithm?**

A: A set of well-defined steps for performing a task or solving a problem

**1.12 Why were computer programming languages invented?**

A: To ease the task of programming. Programs may be written in a programming language,

and then converted to machine language.

**1.13 What is the difference between a high-level language and a low-level language?**

A: A low-level language is close to the level of the computer, and resembles the system’s

numeric machine language. A high-level language is closer to the level of human

readability, and resemble natural languages.

**1.14 What does *portability* mean?**

A: That a program may be written on one type of computer and run on another type

**1.15 Explain the operations carried out by the preprocessor, compiler, and linker.**

A: The preprocessor reads the source file searching for commands that begin with the #

symbol. These are commands that cause the preprocessor to modify the source file in some

way. The compiler translates each source code instruction into the appropriate machine

language instruction, and creates an object file. The linker combines the object file with

necessary library routines.

**1.16 Explain what is stored in a source le, an object le, and an executable le.**

A:

Source file: contains program statements written by the programmer.

Object file: machine language instructions, generated by the compiler translated

from the source file.

Executable file: code ready to run on the computer. Includes the machine language from

an object file, and the necessary code from library routines.

**1.17 What is an integrated development environment**?

A: A programming environment that includes a text editor, compiler, debugger, and other

utilities, integrated into one package

**1.18 Describe the difference between a key word and a programmer-defined identi fier.**

A: A key word has a special purpose, and is defined as part of a programming language. A

programmer-defined identifier is a word or name defined by the programmer.

**1.19 Describe the difference between operators and punctuation symbols.**

A: Operators perform operations on one or more operands. Punctuation symbols mark the

beginning or ending of a statement, or separates items in a list.

**1.20 Describe the difference between a program line and a statement.**

A: A line is a single line as it appears in the body of a program. A statement is a complete

instruction that causes the computer to perform an action.

**1.21 Why are variables called variables ?**

A: Because their contents may be changed.

**1.22 What happens to a variable s current contents when a new value is stored there?**

A: The original value is overwritten  
**1.23 What must take place in a program before a variable is used?**

A: The variable must be defined.

**1.24 What are the three primary activities of a program?**

A: Input, processing, and output

**1.25 What four items should you identify when defining what a program is to do?**

A: The program’s purpose, information to be input, the processing to take place, and the

desired output.

**1.26 What does it mean to visualize a program running ? What is the value of such**

**an activity?**

A: To imagine what the computer screen looks like while the program is running. This helps

define input and output.

**1.27 What is a hierarchy chart?**

A: A chart that depicts each logical step of the program in a hierarchical fashion.

**1.28 Describe the process of desk-checking.**

A: The programmer steps through each statement in the program from beginning to end. The

contents of variables are recorded, and screen output is sketched.

**1.29 Describe what a compiler does with a program s source code.**

A: It translates each source code statement into the appropriate machine language statement..

**1.30 What is a run-time error?**

A: A logical error that occurs while the program is running.

**1.31 Is a syntax error (such as misspelling a key word) found by the compiler or when**

the program is running?

A: By the compiler

**1.32 What is the purpose of testing a program with sample data or input?**

A: To determine if a logical error is present in the program.

**1.33 Briefly describe the difference between procedural and object-oriented programming**

A: Procedural programs are made of procedures, or functions. Object-oriented programs are

centered on objects, which contain both data and the procedures that operate on the data.

**Chapter 2 Checkpoints:**

**2.1 The following C++ program will not compile because the lines have been mixed**

**up.**

**int main()**

**}**

**// A crazy mixed up program**

**return 0;**

**#include <iostream>**

**cout << "In 1492 Columbus sailed the ocean blue.";**

**{**

**using namespace std;**

**When the lines are properly arranged the program should display the following**

**on the screen:**

**In 1492 Columbus sailed the ocean blue.**

**Rearrange the lines in the correct order. Test the program by entering it on the**

**computer, compiling it, and running it.**

**A:** // A crazy mixed up program

#include <iostream>

using namespace std;

int main()

{

cout << "In 1492 Columbus sailed the ocean blue.";

return 0;

}

**2.2 The following C++ program will not compile because the lines have been mixed up.**

**cout << "Success\n";**

**cout << " Success\n\n";**

**int main()**

**cout << "Success";**

**}**

**using namespace std;**

**// It's a mad, mad program**

**#include <iostream>**

**cout << "Success\n";**

**{**

**return 0;**

**When the lines are properly arranged the program should display the following**

**on the screen:**

**Program Output**

**Success**

**Success Success**

**Success**

**Rearrange the lines in the correct order. Test the program by entering it on the**

**computer, compiling it, and running it.**

**A:**

// It's a mad, mad program

#include <iostream>

using namespace std;

int main()

{

cout << "Success\n";

cout << "Success";

cout << " Success\n\n";

cout << "Sucess\n";

return 0;

}

**2.3 Study the following program and show what it will print on the screen.**

**// The Works of Wolfgang**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**cout << "The works of Wolfgang\ninclude the following";**

**cout << "\nThe Turkish March" << endl;**

**cout << "and Symphony No. 40 ";**

**cout << "in G minor." << endl;**

**return 0;**

**}**

**A:**

The works of Wolfgang

include the following

The Turkish March

and Symphony No. 40 in G minor.

**2.4 On paper, write a program that will display your name on the rst line, your street**

**address on the second line, your city, state, and ZIP code on the third line, and**

**your telephone number on the fourth line. Place a comment with today s date at**

**the top of the program. Test your program by entering, compiling, and running it.**

**A:** #include <iostream>

using namespace std;

int main()

{

cout << "Teresa Jones\n";

cout << "127 West 423rd Street\n";

cout << "San Antonio, TX 55555\n";

cout << "555-555-1212\n";

return 0;

}

**2.5 Examine the following program.**

**// This program uses variables and literals.**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int little;**

**int big;**

**little = 2;**

**big = 2000;**

**cout << "The little number is " << little << endl;**

**cout << "The big number is " << big << endl;**

**return 0;**

**}**

**List all the variables and literals that appear in the program.**

**A:** Variables:

little

and

big

. Constants: 2, 2000, “The little number is ”, “The big number is ”

**2.6 What will the following program display on the screen?**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int number;**

**number = 712;**

**cout << "The value is " << "number" << endl;**

**return 0;**

**}**

**2.7 Which of the following are illegal variable names, and why?**

**x**

**99bottles**

**july97**

**theSalesFigureForFiscalYear98**

**r&d**

**grade\_report**

**A:** The value is number

**2.8 Is the variable name Sales the same as sales? Why or why not?**

**A:** No. Variable names are case sensitive.

**2.9 Refer to the data types listed in Table 2-6 for these questions.**

**A) If a variable needs to hold numbers in the range 32 to 6,000, what data type**

**would be best?**

**B) If a variable needs to hold numbers in the range 40,000 to +40,000, what**

**data type would be best?**

**C) Which of the following literals uses more memory? 20 or 20L**

**A:**

A)

short

, or

unsigned

short

.

B)

int

C) They both use the same amount of memory.

**2.10 On any computer, which data type uses more memory, an integer or an unsigned**

**Integer?**

**A**: They both use the same amount of memory.

**2.11 What are the ASCII codes for the following characters? (Refer to Appendix B)**

**C**

**F**

**W**

**A:** 67, 70, 87

**2.12 Which of the following is a character literal?**

**'B'**

**"B"**

**A: ‘B’**

**2.13 Assuming the char data type uses 1 byte of memory, how many bytes do the following**

**literals use?**

**'Q'**

**"Q"**

**"Sales"**

**'\n'**

**A**: ‘Q’ uses one byte

“Q” uses two bytes

“Sales” uses six bytes

‘\n’ uses one byte

**2.14 Write a program that has the following character variables: first, middle, and**

**last. Store your initials in these variables and then display them on the screen.**

**A:** #include <iostream>

using namespace std;

int main()

{first = 'T';

middle = 'E';

last = 'G';

cout << first << " " << middle << " " << last << endl;

return 0;

}

**char first, middle, last;**

**2.15 What is wrong with the following program statement?**

**char letter = "Z";**

**A:** The string constant “Z” is being stored in the character variable

letter

**.**

**2.16 What header le must you include in order to use string objects?**

**A:** The string header file

**2.17 Write a program that stores your name, address, and phone number in three separate**

**string objects. Display the contents of the string objects on the screen.**

**A**: #include <iostream>

#include <string>

using namespace std;

int main()

{

string name = "John Smith";

string address = "224 Maple Street\nClyde, NC 28721";

string phone = "555-5050";

cout << name << endl;

cout << address << endl;

cout << phone << endl << endl;

return 0;

}

**2.18 Yes or No: Is there an unsigned oating point data type? If so, what is it?**

**A:** No

**2.19 How would the following number in scienti c notation be represented in E notation?**

**6.31 \* 1017**

**A:** 6.31E17

**2.20 Write a program that de nes an integer variable named age and a float variable**

**named weight. Store your age and weight, as literals, in the variables. The program**

**should display these values on the screen in a manner similar to the following:**

**Program Output**

**My age is 26 and my weight is 180 pounds.**

**(Feel free to lie to the computer about your age and your weight**

**it ll never know!)**

**A**: #include <iostream>

using namespace std;

int main()

{

int age;

float weight;

age = 26;

weight = 180;

cout << "My age is " << age << endl;

cout << "My weight is " << weight << endl;

return 0;

}

**2.21 Is the following assignment statement valid or invalid? If it is invalid, why?**

**72 = amount;**

**A**: Invalid. The value on the left of the = operator must be an lvalue.

**2.22 How would you consolidate the following de nitions into one statement?**

**int x = 7;**

**int y = 16;**

**int z = 28;**

**A**: int x = 7, y = 16, z = 28;

**2.23 What is wrong with the following program? How would you correct it?**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**number = 62.7;**

**double number;**

**cout << number << endl;**

**return 0;**

**}**

**A:** The variable

number

is assigned a value before it is defined. Correct the program by

moving the statement

number = 62.7;

to the point after the variable declaration. Here is

the corrected program:

#include <iostream>

using namespace std;

int main()

{

double number;

number = 62.7;

cout << number << endl;

return 0;

}

**2.24 Is the following an example of integer division or oating-point division? What**

**value will be stored in portion?**

**portion = 70 / 3;**

**A:** Integer division. The value 23 will be stored in

portion

**.**

**2.25 Write statements using the const quali er to create named constants for the following**

**literal values:**

**Literal Value Description**

**2.71828 Euler s number (known in mathematics as *e*)**

**5.256E5 Number of minutes in a year**

**32.2 The gravitational acceleration constant (in feet per second2)**

**9.8 The gravitational acceleration constant (in meters per second2)**

**1609 Number of meters in a mile**

**A:** const float E = 2.71828;

const int MINUTES\_IN\_A\_YEAR = 5.256E5;

const float G\_FEET = 32.2;

const float G\_METERS = 9.8;

const int METERS\_IN\_A\_MILE = 1609

**Chapter 3 Checkpoints**

**3.1 What header le must be included in programs using cin?**

A: iostream

**3.2 TRUE or FALSE: cin requires the user to press the [Enter] key when nished**

**entering data.**

A: True

**3.3 Assume value is an integer variable. If the user enters 3.14 in response to the following**

**programming statement, what will be stored in value?**

**cin >> value;**

**A) 3.14**

**B) 3**

**C) 0**

**D) Nothing. An error message is displayed**.

A: B

**3.4 A program has the following variable de nitions.**

**long miles;**

**int feet;**

**float inches;**

**Write one cin statement that reads a value into each of these variables.**

A: cin >> miles >> feet >> inches

**3.5 The following program will run, but the user will have dif culty understanding**

**what to do. How would you improve the program?**

**// This program multiplies two numbers and displays the result.**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**double first, second, product;**

**cin >> first >> second;**

**product = first \* second;**

**cout << product;**

**return 0;**

**}**

A: Include one or more cout statements explaining what values the user should enter.

**3.6 Complete the following program skeleton so it asks for the user s weight (in**

**pounds) and displays the equivalent weight in kilograms.**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**double pounds, kilograms;**

**// Write code here that prompts the user**

**// to enter his or her weight and reads**

**// the input into the pounds variable.**

**// The following line does the conversion.**

**kilograms = pounds / 2.2;**

**// Write code here that displays the user's weight**

**// in kilograms.**

**return 0;**

**}**

A:

#include <iostream>

using namespace std;

int main()

{

double pounds, kilograms;

cout << "Enter your weight in pounds: ";

cin >> pounds;

// The following line does the conversion.

// One kilogram weighs 2.2 pounds.

kilograms = pounds / 2.2;

cout << "Your weight in kilograms is ";

cout << kilograms << endl;

return 0;

}:

**3.7 Complete the table below by writing the value of each expression in the Value**

**column.**

A:

*Value*

21

2

31

5

24

2

69

0

30

**Program 3-6**

1 // This program calculates the average

2 // of three test scores.

3 #include <iostream>

4 #include <cmath>

5 using namespace std;

6

7 int main()

8 {

9 double test1, test2, test3; // To hold the scores

10 double average; // To hold the average

11

12 // Get the three test scores.

13 cout << "Enter the first test score: ";

14 cin >> test1;

15 cout << "Enter the second test score: ";

16 cin >> test2;

17 cout << "Enter the third test score: ";

18 cin >> test3;

19

20 // Calculate the average of the scores.

21 average = (test1 + test2 + test3) / 3.0;

22

23 // Display the average.

24 cout << "The average score is: " << average << endl;

25 return 0;

26 }

**Program Output with Example Input Shown in Bold**

Enter the first test score: **90 [Enter]**

Enter the second test score: **80 [Enter]**

Enter the third test score: **100 [Enter]**

The average score is 90

**Expression Value**

6 + 3 \* 5

12 / 2 - 4

9 + 14 \* 2 - 6

5 + 19 % 3 - 1

(6 + 2) \* 3

14 / (11 - 4)

9 + 12 \* (8 - 3)

(6 + 17) % 2 - 1

(9 - 3) \* (6 + 9) / 3

**3.8 Write C++ expressions for the following algebraic expressions:**

A: y = 6 \* x;

a = 2 \* b + 4 \* c;

y = x \* x;

or

y = pow(x, 2.0);

g = (x + 2) / (z \* z);

or

g = (x + 2.0) / pow(z, 2.0);

y = (x \* x) / (z \* z);

or

y = pow(x, 2.0) / pow (z, 2.0);

**3.9 Study the following program and complete the table.**

**#include <iostream>**

**#include <cmath>**

**using namespace std;**

**int main()**

**{**

**double value1, value2, value3;**

**cout << "Enter a number: ";**

**cin >> value1;**

**value2 = 2 \* pow(value1, 2.0);**

**value3 = 3 + value2 / 2 - 1;**

**cout << value3 << endl;**

**return 0;**

**}**

A:

*If the user enters… The program displays…*

2 6

5 27

4.3 20.49

6 38

**3.10 Complete the following program skeleton so it displays the volume of a cylindrical**

**fuel tank. The formula for the volume of a cylinder is**

**Volume = *r*2*h***

**where**

**is 3.14159**

***r* is the radius of the tank**

***h* is the height of the tank**

**#include <iostream>**

**#include <cmath>**

**using namespace std;**

**If the User Enters**

**The Program Will Display What Number**

**(Stored in value3)?**

**2**

**5**

**4.3**

**6**

***y* = 6*x***

***a* = 2*b* + 4*c***

***y x*2**

**=**

***g***

***x* + 2**

***z*2**

**= ------------**

***y***

***x*2**

**int main()**

**{**

**double volume, radius, height;**

**cout << "This program will tell you the volume of\n";**

**cout << "a cylinder-shaped fuel tank.\n";**

**cout << "How tall is the tank? ";**

**cin >> height;**

**cout << "What is the radius of the tank? ";**

**cin >> radius;**

**// You must complete the program.**

**}**

A**:**

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

double volume, radius, height;

cout << "This program will tell you the volume of\n";

cout << "a cylinder-shaped fuel tank.\n";

cout << "How tall is the tank? ";

cin >> height;

cout << "What is the radius of the tank? ";

cin >> radius;

volume = 3.14159 \* pow(radius, 2.0) \* height;

cout << "The volume of the tank is " << volume << endl;

return 0;

}

**3.11 Assume the following variable de nitions:**

**int a = 5, b = 12;**

**double x = 3.4, z = 9.1;**

**What are the values of the following expressions?**

**A) b / a**

**B) x \* a**

**C) static\_cast<double>(b / a)**

**D) static\_cast<double>(b) / a**

**E) b / static\_cast<double>(a)**

**F) static\_cast<double>(b) / static\_cast<double>(a)**

**G) b / static\_cast<int>(x)**

**H) static\_cast<int>(x) \* static\_cast<int>(z)**

**I) static\_cast<int>(x \* z)**

**J) static\_cast<double>(static\_cast<int>(x) \***

**static\_cast<int>(z))**

A:

A) 2

B) 17.0

C) 2.0

D) 2.4

E) 2.4

F) 2.4

G) 4

H) 27

I) 30

J) 27.0

**3.12 Complete the following program skeleton so it asks the user to enter a character.**

**Store the character in the variable letter. Use a type cast expression with the variable**

**in a cout statement to display the character s ASCII code on the screen.**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**char letter;**

**// Finish this program**

**// as specified above.**

**return 0;**

**}**

A:

#include <iostream>

using namespace std;

int main()

{

char letter;

cout << "Enter a character: ":

cin >> letter;

cout << "The ASCII code for " << letter;

cout << " is " << static\_cast<int>(letter) << endl;

return 0;

}

**3.13 What will the following program display?**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int integer1, integer2;**

**double result;**

**integer1 = 19;**

**integer2 = 2;**

**result = integer1 / integer2;**

**cout << result << endl;**

**result = static\_cast<double>(integer1) / integer2;**

**cout << result << endl;**

**result = static\_cast<double>(integer1 / integer2);**

**cout << result << endl;**

**return 0;**

**}**

A:

9

9.5

9

**3.14 Write a multiple assignment statement that assigns 0 to the variables total,**

**subtotal, tax, and shipping.**

A: total = subtotal = tax = shipping = 0;

**3.15 Write statements using combined assignment operators to perform the following:**

**A) Add 6 to x.**

**B) Subtract 4 from amount.**

**C) Multiply y by 4.**

**D) Divide total by 27.**

**E) Store in x the remainder of x divided by 7.**

**F) Add y \* 5 to x.**

**G) Subtract discount times 4 from total.**

**H) Multiply increase by salesRep times 5.**

**I) Divide profit by shares minus 1000.**

A:

A)

x += 6;

B)

amount -= 4;

C)

y \*= 4;

D)

total /= 27;

E)

x %= 7;

F)

x += (y \* 5);

G)

total -= (discount \* 4);

H)

increase \*= (salesRep \* 5);

I)

profit /= (shares – 1000);

**3.16 What will the following program display?**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int unus, duo, tres;**

**unus = duo = tres = 5;**

**unus += 4;**

**duo \*= 2;**

**tres -= 4;**

**unus /= 3;**

**duo += tres;**

**cout << unus << endl;**

**cout << duo << endl;**

**cout << tres << endl;**

**return 0;**

**}**

A:

3

11

1

**3.17 Write cout statements with stream manipulators that perform the following:**

**A) Display the number 34.789 in a eld of nine spaces with two decimal places**

**of precision.**

**B) Display the number 7.0 in a eld of ve spaces with three decimal places of**

**precision.**

**The decimal point and any trailing zeroes should be displayed.**

**C) Display the number 5.789e+12 in xed point notation.**

**D) Display the number 67 left justi ed in a eld of seven spaces.**

A:

A)

cout << setw(9) << fixed << setprecision(2) << 34.789;

B)

cout << setw(5) << fixed << setprecision(3) << 7.0;

C)

cout << fixed << 5.789e12;

D)

cout << left << setw(7) << 67;

**3.18 The following program will not compile because the lines have been mixed up.**

**#include <iomanip>**

**}**

**cout << person << endl;**

**string person = "Wolfgang Smith";**

**int main()**

**cout << person << endl;**

**{**

**#include <iostream>**

**return 0;**

**cout << left;**

**using namespace std;**

**cout << setw(20);**

**cout << right;**

**When the lines are properly arranged the program should display the following:**

**Wolfgang Smith**

**Wolfgang Smith**

**Rearrange the lines in the correct order. Test the program by entering it on the**

**computer, compiling it, and running it.**

A:

#include <iostream>

#include <string>

using namespace std;

int main()

{

string person = "Wolfgang Smith";

cout << right;

cout << setw(20);

cout << person << endl;

cout << left;

cout << person << endl;

return 0;

}

**3.19 The following program skeleton asks for an angle in degrees and converts it to**

**radians. The formatting of the nal output is left to you.**

**#include <iostream>**

**#include <iomanip>**

**using namespace std;**

**int main()**

**{**

**const double PI = 3.14159;**

**double degrees, radians;**

**cout << "Enter an angle in degrees and I will convert it\n";**

**cout << "to radians for you: ";**

**cin >> degrees;**

**radians = degrees \* PI / 180;**

**// Display the value in radians left justified, in fixed**

**// point notation, with 4 places of precision, in a field**

**// 5 spaces wide, making sure the decimal point is always**

**// displayed.**

**return 0;**

**}**

A;

#include <iostream>

#include <iomanip>

using namespace std;

int main()

{

const double PI = 3.14159;

double degrees, radians;

cout << "Enter an angle in degrees and I will convert it\n";

cout << "to radians for you: ";

cin >> degrees;

radians = degrees \* PI / 180;

cout << degrees << " degrees is equal to ";

cout << setw(5) << left << fixed << showpoint

<< setprecision(4) << radians << " radians.\n";

return 0;

}

**3.20 Write a short description of each of the following functions:**

**cos log sin**

**exp log10 sqrt**

**fmod pow tan**

A:

cos

: Returns the cosine of the argument.

exp

: Returns the exponential function of the argument.

fmod

: Returns the remainder of the first argument divided by the second argument.

log

: Returns the natural logarithm of the argument.

log10

: Returns the base-10 logarithm of the argument.

pow

: Returns the value of the first argument raised to the power of the second

argument.

sin

: Returns the sine of the argument.

sqrt

: Returns the square root of the argument.

tan

: Returns the tangent of the argument

**3.21 Assume the variables angle1 and angle2 hold angles stored in radians. Write a**

**statement that adds the sine of angle1 to the cosine of angle2, and stores the**

**result in the variable x.**

A: x = sin(angle1) + cos(angle2);

**3.22 To nd the cube root (the third root) of a number, raise it to the power of 1**

**3. To**

**nd the fourth root of a number, raise it to the power of 1**

**4. Write a statement**

**that will nd the fth root of the variable x and store the result in the variable y.**

A: y = pow(x, 0.2); // 0.2 is equal to 1/5

**3.23 The cosecant of the angle *a* is**

**Write a statement that calculates the cosecant of the angle stored in the variable a,**

**and stores it in the variable y.**

A: y = 1 / sin(a);

**Chapter 1: Key Topics:**

1.1 Why Program?

1.2 Computer Systems: Hardware

and Software

1.3 Programs and Programming

Languages

1.4 What Is a Program Made of?

1.5 Input, Processing, and Output

1.6 The Programming Process

1.7 Procedural and Object-Oriented

Programming

**Chapter 2: Key Topics:**

2.1 The Parts of a C++ Program

2.2 The cout Object

2.3 The #include Directive

2.4 Variables and Literals

2.5 Identi ers

2.6 Integer Data Types

2.7 The Char Data Type

2.8 The String Class

2.9 Floating-Point Data Types

2.10 The bool Data Type

2.11 Determining the Size of a Data Type

2.12 Variable Assignments and Initialization

2.13 Scope

2.14 Arithmetic Operators

2.15 Comments

2.16 Named Constants

2.17 Programming Style

2.18 If You Plan to Continue

in Computer Science: Standard

and Prestandard C++

**Chapter 3: Key Topics**:

3.1 The cin Object

3.2 Mathematical Expressions

3.3 When You Mix Apples and Oranges: Type Conversion

3.4 Over ow and Under ow

3.5 Type Casting

3.6 Multiple Assignment and Combined Assignment

3.7 Formatting Output

3.8 Working with Characters and string Objects

3.9 More Mathematical Library Functions

3.10 Focus on Debugging: Hand Tracing a Program

3.11 Focus on Problem Solving:

A Case Study