2.1 Introduction

(No questions.)

2.2 A Simple Program: Printing a Line of Text

2.1 Which of the following must every C program have?
(a) `main`
(b) `#include`
(c) `/*`
(d) `<stdio.h>`
ANS: (a)

2.2 Every statement in C must end with a
(a) period (`.`)
(b) semicolon (`;`)
(c) colon (`:`)
(d) backslash (`/`)
ANS: (b)

2.3 Which of the following is *not* a valid escape sequence?
(a) `\n`
(b) `\\`
(c) `\~`
(d) `\"`
ANS: (c)

2.4 Which statement about comments is *false*?
a) Comments begin and end with `/*` and `*/`, respectively.
b) Programmers insert comments to document programs and improve program readability.
c) Comments do not cause any machine language object code to be generated.
d) Lengthy comments can cause poor execution-time performance.
ANS: (d)

2.5 Lines beginning with a `#` are processed
a) at execution time.
b) at compile time.
c) at preprocessor time.
d) at postprocessor time.
ANS: (c)

2.6 Which of the following statements about the inclusion of `<stdio.h>` is *false*?

a) It is required.
b) This header file contains information and declarations used by the compiler when compiling standard input/output library functions such as `printf`.
c) This header file contains information that helps the compiler determine if calls to library functions have been made correctly.
d) This header helps locate bugs in your program at compile time, rather than at execution time (when errors are usually more costly to correct).
ANS: (a)

2.7 In the line

```
int main()
```

the parentheses indicate that `main` is a program building block called a
a) module
b) statement
c) directive
d) function
ANS: (d)

2.8 The pair of braces that delineate the body of main *and* the portion of the program between these braces is called a _____.
a) function
b) block
c) statement
d) header
ANS: (b)

2.9 Which of the following is *not* a synonym for a C string?
a) message
b) character string
c) character
d) literal
ANS: (c)

2.10 The following line is most properly an example of a _____.

```
puts( "Welcome to C!" );
```

a) function
b) block
c) statement
d) header
ANS: (c)

2.11 In a `printf`, a backslash is printed by enclosing in quotes
a) `\`
b) `\\`
c) `/\`
d) `//`
ANS: (b)

2.12 A linked program is often called a(n) _____.
a) chain
b) library
c) object
d) executable
ANS: (d)

2.13 The escape sequence for horizontal tab is _____.
a) `\tab`
b) `\t`
c) `\horizontaltab`
d) `\T`
ANS: (b)

2.3 Another Simple Program: Adding Two Integers

2.14 Which of the following is *not* a valid integer value?
(a) `-3`
(b) `0`
(c) `2134859`
(d) `1.1`
ANS: (d)

2.15 Which of the following is an *invalid* identifier (variable name)?
(a) `_Test`
(b) `TEST`
(c) `5test`
(d) `test1`
ANS: (c)

2.16 Which statement prints “hi” on the screen?
(a) `puts("hi");`

(b) `put "hi";`
(c) `puts "hi";`
(d) none of the above
ANS: (a)

2.17 The _____ sign is also known as the _____ operator.
(a) `+`, assignment
(b) `=`, assignment
(c) `*`, stream manipulator
(d) `&`, stream insertion
ANS: (b)

2.18 A(n) _____ is a location in the computer's memory where a value can be stored for use by a program.
a) unknown
b) name
c) variable
d) declaration
ANS: (c)

2.19 Which statement is *false*.
a) Variables may be defined anywhere in the body of `main`.
b) All variables must be defined before they are used.
c) All variable definitions must include the name and data type of each variable.
d) Several variables of the same data type may be defined in one definition.
ANS: (a)

2.20 Which of these is *not* a valid identifier?
a) `a_valid_identifier`
b) `a1_valid_identifier`
c) `a_valid_identifier_`
d) `1_valid_identifier`
ANS: (d)

2.21 Which of the following statements is *false*?
a) C is case sensitive.
b) Uppercase and lowercase letters are different in C.
c) identifier and IdEnTiFiEr are identical identifiers in C.
d) Identifiers can be of any length
ANS: (c)

2.22 Which of the following multiple word variable names does *not* conform to the good programming practices in the text?
a) `multiple_word_variable_name`

b) `multipleWordVariableName`
c) `multiplewordvariablename`
d) `aReallyReallyLongMultipleWordVa`
ANS: (c)

2.23 The address operator is
a) `&&`
b) `%`
c) `@`
d) `&`
ANS: (d)

2.24 Which statement is *false*?
a) in the statement

```
sum = integer1 + integer2;
```

both = and + are binary operators.
b) The statement in part a) is an example of an assignment statement.
c) The spaces around each of the binary operators in the statement of part a) are required.
d) In part a), the = operator's two operands are `sum` and the value of the expression `integer1 + integer2`.
ANS: (c)

2.25 Which of the following is *false*?
a) Each variable being input in a `scanf` statement is generally preceded by an `&`.
b) Each variable being output in a `printf` statement is generally not preceded by an `&`.
c) In a `printf` statement, the comma that separates the format control string from the expressions to be printed is placed inside the format control string.
d) Calculations can be performed inside `printf` statements.
ANS: (c)

2.4 Memory Concepts

2.26 Variable names actually correspond to _____.
(a) locations in the computer's memory
(b) operators
(c) integers
(d) data types
ANS: (a)

2.27 When a number gets assigned to a variable that already has a value _____.

(a) the new number overwrites the previous value at that memory location
(b) the new number gets assigned to a neighboring memory location
(c) the computer issues an error
(d) the new value is destroyed and the old value remains
ANS: (a)

2.28 Every variable has all the attributes below, except
a) name
b) value
c) alias
d) type
ANS: (c)

2.29 Which of the following is *false*?
a) Reading a value into a memory location destroys the previous value.
b) Reading a value out of a memory location destroys that value.
c) `sum = integer1 + integer2;` involves destructive read-in.
d) The statement in part c) also involves nondestructive read-out.
ANS: (b)

2.5 Arithmetic in C

2.30 Which operation will find the remainder when 15 is divided by 6?
(a) `15 / 6`
(b) `15 % 6`
(c) `15 ^ 6`
(d) `15 * 6`
ANS: (b)

2.31 Evaluate the expression

```
3 * 4 % 6 + 4 * 5
```

(a) 20
(b) 26
(c) 12
(d) 32
ANS: (a)

2.32 Which statement is *false*?
a) In algebra, we write *ab* to multiply *a* times *b*.
b) In C, we write `ab` to multiply `a` times `b`.
c) In C, the remainder operator is `%`.

d) In C, integer division yields an integer result.
ANS: (b)

2.33 Which statement about C arithmetic is *false*?
a) 6 / 3 yields 2
b) 5 / 2 yields 2.5
c) 7 % 3 yields 1
d) 6 % 3 yields 0
ANS: (b)

2.34 `a * ( b + c )` may also be written in C as
a) `ab + ac`
b) `(a * b ) + c`
c) `a * b + c`
d) `a * b + a * c`
ANS: (d)

2.35 Which statement about precedence is *false*?
a) Parentheses may be used to force the order of evaluation to occur in any sequence desired by the programmer.
b) Nested, or embedded parentheses are evaluated last.
c) Multiplication has a higher precedence than addition.
d) Subtraction has a lower precedence than division.
ANS: (b)

2.36 Which expression is *true*?
a) The expression `a * (b + c) + c * (d + e)` contains nested parentheses.
b) The expression `y = a * x * x + b * x + c` does exponentiation *without* an exponentiation operator.
c) The C standard library provides function `power` to perform exponentiation.
d) When we say evaluation of an expression proceeds from left to right we are referring to the additivity of the operators.
ANS: (b)

2.6 Decision Making: Equality and Relational Operators

2.37 C's `if` statement executes the statement inside its body if a specified _____ is _____.
(a) condition, `true`
(b) condition, `false`
(c) equality operator, `true`
(d) relational operator, `true`

ANS: (a)

2.38 Which of the following is an equality operator?
(a) ==
(b) =
(c) >
(d) >=
ANS: (a)

2.39 Which statement is *false*?
a) Executable C statements either perform actions or make decisions.
b) If the condition in an `if` statement is met, the statement in the body of the if statement is executed.
c) All the relational operators have the same level of precedence.
d) The equality operators have a higher level of precedence than the relational operators.
ANS: (d)

2.40 Which statement is false?
a) Whitespace characters such as tabs, newlines and spaces are generally ignored by the C compiler.
b) The statements in an `if` statement must be indented.
c) Placing a blank line before and after every control structure can improve program readability.
d) There can be (but should not be) more than one statement per line.
ANS: (b)

2.41 Which statement is *false*?
a) It is *not* correct to split an identifier with a space, a tab or a newline.
b) Statements and comments may be split over several lines.
c) The equals sign (=) is not an operator.
d) A good programming practice is to break a line after a comma in a lengthy comma-separated list.
ANS: (c)

2.42 Which of the following is *not* a keyword?
a) `int`
b) `return`
c) `if`
d) `main`
ANS: (d)

2.43 Which statement is *false*?
a) The assignment operator associates from left to right.
b) The arithmetic operators associate from left to right.

c) The equality operators associate from left to right.
d) The relational operators associate from left to right.
ANS: (a)

2.44 The order in which statements are _____ is called flow of control.
a) entered in a source file
b) preprocessed
c) compiled
d) executed
ANS: (d)

2.7 Secure C Programming

2.45 Which of the following statements is true in secure C programming?
(a) You should avoid using `printf` to display a single string argument.
(b) You should always use `printf` to display a single string argument.
(c) You should always use `puts` to display a single string argument.
(d) None of the above.
ANS: (a)

2.46 Which of the following statements should be used in secure C programming to display the string "Welcome" *not* followed by a newline character?
(a) `printf( "Welcome" );`
(b) `puts( "Welcome" );`
(c) `printf( "%s", "Welcome" );`
(d) None of the above.
ANS: (c)