060010101 – Fundamentals of Programming
Question Bank

UNIT: Introduction of Computers, Logic and Structure

Short Questions

1. Which are the two major components of any computer system?
2. Define: (1)Hardware (2)Software
3. List out the operations accomplished by hardware and software together.
4. List out any five programming languages.
5. Which are the two categories of computer storage?
6. Give the example of internal storage and external storage.
7. List out steps of programming process.
8. Explain in brief any one step of programming process.
9. Which are the tools most commonly used for planning the program's logic?
10. Give the name of the smallest unit in the data hierarchy.
11. What do you mean by flowchart and pseudocode?
12. Draw the following symbols used in flowchart: Start/Stop/Terminal, Input, Processing, Decision, Output, and Connector.
13. List any two naming rules for variables that should be followed when designing program logic.
14. List the naming conventions for variables in C.
15. What do you mean by infinite loop?
16. What is sentinel value?
17. What is the difference between on-page and off-page connector? Draw the symbols for both.
18. Which are the two basic data types that the computers deal with?
19. What does a variable's data type describe?
20. List the two major programming techniques.
21. Match the definition with the appropriate term.

1.	Computer system devices	a.	Compiler
2.	Another word for programs	b.	Syntax
3.	Language rules	c.	Logic
4.	Order of instructions	d.	Hardware
5.	Language translator	e.	Software

22. Explain priming read in brief.
23. Which three structures are better and for what reasons?
24. Draw the figure for the selection structure.
25. What do you mean by "eof"?
26. Give any one example of single alternative selection.
27. Which of the following names seem like good variable name to you? If a name does not seem like a good variable name, explain why not?
    a) c
    b) cost
    c) costAmount

d) cost amount
e) cstofdngbsns
f) costOf Doing BusinessThis Year
g) cost2007

28. What do you mean by Programming Language?

Long Questions

1. Write a short note on operations performed by computer hardware and software with example.
2. Explain the software used to translate the program into machine language.
3. Explain programming process in brief.
4. Write a short note on the symbols used to draw a flowchart.
5. How to manage the large flow chart explain with example.
6. Write a pseudo code to find the area of the rectangle.
7. Draw a flowchart or write pseudo code to represent the logic of a program that allows the user to enter a value. Program doubles that value and outputs the result.
8. Draw a flowchart or write pseudo code to represent the logic of a program that allows the user to enter two values and outputs the sum of those values.
9. Find the bug:
   This pseudocode segment is intended to describe computing your average score of two classroom tests.
   ```
   input midtermGrade
   input finalGrade
   average = (inputGrade + final) / 3
   print average
   ```
10. Explain any two naming rules for variables that should be followed when designing program logic.
11. Explain Data Hierarchy.
12. Explain the naming convention for variables in C.
13. Write a short note on Data Types.
14. Write a note on Spaghetti code.
15. Explain priming read in brief.
16. List the three basic structures. Explain loop structure in detail.
17. Explain Selection structure with appropriate example.
18. Explain sentinel values using appropriate example.
19. Explain C program structure.
20. Explain printf() function using example.
21. Explain scanf() function using example.
22. Differentiate between variable and constant.
23. Write a short note on variable.
24. Find the bug:
    This pseudocode segment is intended to describe determining whether you have passed or failed a course based on the average score of two classroom tests:

```
input midtermGrade
input finalGrade
average=(inputGrade+finalGrade)/2
print avg
if average >=60 then
    print "Pass"
endif
else
    print "Fail"
```

25. Write a note on arithmetic operators of C.

Multiple Choice Questions

1. Which are the major components of any computer system?
   a. Hardware and software
   b. input, processing, and output
   c. sequence and looping
   d. spreadsheets, word processing, and data communications
2. The major computer operations include _____.
   a. hardware and software
   b. input, processing, and output
   c. sequence and looping
   d. spreadsheets, word processing, and data communications
3. Another term meaning "computer instructions" is _____.
   a. hardware
   b. software
   c. queries
   d. data
4. The most important task of a compiler or interpreter is to _____.
   a. create the rules for a programming language
   b. translate English statements into a language such as Java
   c. translate programming language statements into machine language
   d. execute machine language programs to perform useful tasks

3. Which of the following is temporary, internal storage?
   a. CPU
   b. hard disk
   c. keyboard
   d. memory

4. Which of the following pairs of steps in the programming process is in the correct order?
   a. code the program, plan the logic
   b. test the program, translate it into machine language
   c. put the program into production, understand the problem
   d. code the program, translate it into machine language

5. The programmer's most important task before planning the logic of a program is to _____.
   a. decide which programming language to use
   b. code the problem

c. train the users of the program
d. understand the problem

6. Writing a program in a language such as C++ or Java is known as _____ the program.
   a. translating
   b. coding
   c. interpreting
   d. compiling

7. An English-like programming language such as Java or Visual Basic is a _____ programming language.
   a. machine-level
   b. low-level
   c. high-level
   d. binary-level

8. Which of the following is an example of a syntax error?
   a. producing output before accepting input
   b. subtracting when you meant to add
   c. misspelling a programming language word
   d. all of the above

9. Which of the following is an example of a logical error?
   a. performing arithmetic with a value before inputting it
   b. accepting two input values when a program requires only one
   c. dividing by 3 when you meant to divide by 30
   d. all of the above

10. In a flowchart, a rectangle represents _____.
    a. input
    b. a sentinel
    c. a question
    d. processing

11. In flowcharts, the decision symbol is a _____.
    a. parallelogram
    b. rectangle
    c. lozenge
    d. diamond

12. The term "eof" represents _____.
    a. a standard input device
    b. a generic sentinel value
    c. a condition in which no more memory is available for storage
    d. the logical flow in a program

13. When you use an IDE, as opposed to a simple text editor, to develop a program, _____.
    a. the logic is more complicated
    b. the logic is simpler
    c. the syntax is different
    d. some help is provided

14. When you write a program that will run in a GUI environment, as opposed to a command-line environment, _____.
    a. the logic is very different
    b. some syntax is different
    c. you do not need to plan the logic
    d. users are more confused

15. With object-oriented programming, as opposed to procedural programming, _____.
    a. the programmer's focus differs
    b. you cannot use some languages, such as Java
    c. you do not accept input
    d. you do not code calculations; they are created automatically

16. The three structures of structured programming are _____.
    a. sequence, order, and process
    b. selection, loop, and iteration
    c. sequence, selection, and loop
    d. if, else, and then

17. A sequence structure can contain _____.
    a. any number of tasks
    b. exactly three tasks
    c. no more than three tasks
    d. only one task

18. Which of the following is not another term for a selection structure?
    a. decision structure
    b. if-then-else structure
    c. dual-alternative if structure
    d. loop structure

19. The structure in which you ask a question, and, depending on the answer, take some action and then ask the question again, can be called all of the following except _____.
    a. iteration
    b. loop
    c. repetition
    d. if-then-else

20. Attaching structures end to end is called _____.
    a. stacking
    b. untangling
    c. building
    d. nesting

21. The statement if age >= 65 then seniorDiscount = "yes" is an example of a _____.
    a. sequence
    b. loop
    c. dual-alternative selection
    d. single-alternative selection

22. The statement while temperature remains below 60, leave the furnace on is an example of a _____.
    a. sequence
    b. loop
    c. dual-alternative selection
    d. single-alternative selection

23. The statement if age < 13 then movieTicket = 4.00 else movieTicket = 8.50 is an example of a _____.
    a. sequence
    b. loop
    c. dual-alternative selection
    d. single-alternative selection

24. Which of the following attributes do all three basic structures share?
    a. Their flowcharts all contain exactly three processing symbols.
    b. They all have one entry and one exit point.
    c. They all contain a decision.
    d. They all begin with a process.

26. When you input data in a loop within a program, the input statement that precedes the loop _____.
    a. is the only part of the program allowed to be unstructured
    b. cannot result in eof
    c. is called a priming input
    d. executes hundreds or even thousands of times in most business programs

27. Which of the following is acceptable in a structured program?
    a. placing a sequence within the true half of a dual-alternative decision
    b. placing a decision within a loop
    c. placing a loop within one of the steps in a sequence
    d. All of these are acceptable.

28. In a selection structure, the structure-controlling question is _____.
    a. asked once at the beginning of the structure
    b. asked once at the end of the structure
    c. asked repeatedly until it is false
    d. asked repeatedly until it is true

29. In a loop, the structure-controlling question is _____.
    a. asked exactly once
    b. never asked more than once
    c. asked at both before and after the loop body executes
    d. asked only if it is true, and not asked if it is false

30. Which of the following is not a reason for enforcing structure rules in computer programs?

a. Structured programs are clearer to understand than unstructured ones.
b. Other professional programmers will expect programs to be structured.
c. Structured programs usually are shorter than unstructured ones.
d. Structured programs can be broken down into modules easily.

31. Which of the following is not a benefit of modularizing programs?

a. Modular programs are easier to read and understand than non modular ones.
b. If you use modules, you can ignore the rules of structure.
c. Modular components are reusable in other programs.
d. Multiple programmers can work on different modules at the same time.

32. Which of the following is true of structured logic?

a. You can use structured logic with newer programming languages, such as Java and C#, but not with older ones.
b. Any task can be described using some combination of the three structures.
c. Structured programs require that you break the code into easy-to-handle modules that each contains no more than five actions.
d. All of these are true.

33. The two broadest types of data are _____.

a. internal and external
b. volatile and constant
c. text and numeric
d. permanent and temporary

34. Which of the following is not a computer language?

a. Assembly/symbolic language
b. Binary language
c. High-level languages
d. Machine language
e. Natural language

Fill in the blanks

1. A programming language's rules are its _____.
2. The two most commonly used tools for planning a program's logic are _____.
3. Computer programs also are known as _____.
4. Visual Basic, C++, and Java are all examples of computer _____.
5. The value stored in an uninitialized variable is _____.
6. The popular name for logically snarled program statements is _____ code.

7. _____ is a statement that reads the first input whether it is a single data item or a record.
8. A group of statements that executes as a unit is a _____.
9. Placing a structure within another structure is called _____ the structures.
10. A computer program must be free of ____ errors before you can execute it.
11. A loop that runs forever is called ____ loop.
12. A code stored in a file that marks the end of the data is called a(n) ____ marker.
13. The smallest integer value that can be stored in integer variable is ____.
14. ____ keyword is used to define real number.
15. Format specifier for double is ____.
16. The set of instruction that programmer write in programming language is called _____.
17. After compilation, ____ file is generated in C.
18. The process of finding and correcting error is called ____.
19. ____ contains the programmer's original program code.
20. Every program statement in C program must end with a ____.
21. ____function is used to display the output on the screen.

True/False

1. C is a structured programming language.
2. Every program in C ends with END keyword.
3. Every statement is terminated by a colon (':') in C.
4. The purpose of a header file, such as stdio.h, is to store a program's source code.
5. Any valid printable ASCII character is used in identifier.
6. You can use main() two times in C program.
7. printf() can be used to read value from user in C.
8. Attaching structure at the end of another structure is called nested structure.
9. Defining structure within the structure is called nested structure.
10. Declaration of a variable is possible at anywhere in C program.
11. C is a case sensitive language.
12. Sentinel value is the value which is used to give a signal to stop the execution of the

program.

13. *name is a valid variable name.
14. a=a*2 // here 2 is a literal
15. a+b=c is the correct assignment statement.
16. IF is a valid identifier.
17. Dennis Ritchie designed C at Bell Laboratories.
18. C program must have main() function.
19. You can define constant using: #define PI 3.14
20. You can define constant using: const flaot pi=3.14;
21. Literal is also one kind of constant.
22. Comments cause the computer to print the text enclosed between /* and */ when executed.
23. All variables must be given a type when it is declared.

UNIT: The Program Planning Process and Making Decisions

Short Questions

1. What do you mean by end users and programmers?
2. Which are the two categories into which program documentation?
3. List the two most common types of output.
4. Write down full form of GUI.
5. List the guidelines that guide how to prepare input for the program.
6. List at least four aspects of program design that should be considered while program designing.
7. Which are the tactics for designing clear statements?
8. Draw the flowchart segment that represent: (1)a dual-alternative selection structure (2) a single-alternative selection structure
9. Give alternative names for following structures:
   (1)a dual-alternative selection structure
   (2) a single-alternative selection structure
10. What is Boolean Expression?
11. List comparison (relational or relational comparison) operators of C language.
12. List logical operators of C language.
13. What do you mean by truth table?
14. Draw truth table for
    (1) AND
    (2) OR
    (3) NOT
15. Write down general rules for *AND* and *OR* decisions.
16. What is short-circuit evaluation?
17. Which operators take precedence when combining AND and OR operators?
18. Give the name of a problem-analysis tool.
19. List four parts of a decision table.
20. Describe the rules for Evaluation of Expression in C.
21. When is the default keyword useful in a switch-case construct?
22. What is the error in each of the following statements?
    a. If(m==10 | n!=0)

```
        printf("OK");
b.  If(a=<10)
        printf("Jump");
```

Long Questions

1. Write a short note on program documentation.
2. Explain guidelines for naming identifiers in program design.
3. Write a note on tactics for designing clear statements.
4. Explain the use of temporary variables with an example.
5. Differentiate local variable, global variable, and constants using examples.
6. Explain relational operators of C language with example.
7. Explain logical operators of C language with example.
8. Explain precedence of operators when combining AND and OR selections.
9. Write a note on precedence of arithmetic operators in C.
10. Explain the use of case structure.
11. Write a note on decision tables.
12. Write a short note on followings:
    a. Assignment operators
    b. Increment and Decrement operators
    c. Conditional operator
    d. Bitwise operators
    e. Special operators (comma operator and sizeof operator)
13. Explain implicit and explicit type conversion.
14. Explain following functions of math.h library:
    a. ceil(x)
    b. exp(x)
    c. floor(x)
    d. pow(x,y)
    e. sqrt(x)
15. Explain different forms of *if* statement in C.
16. Explain goto statement in C language.
17. Write a pseudocode and draw a flowchart to find the sum of all integers greater than 100 and less than 200 that are divisible by 7.
18. Compare if and switch statement.

Multiple Choice Questions

1. A computer system's standard input device is most often a
   a. Mouse
   b. Floppy disk
   c. Keyboard
   d. Compact disk

2. In most common programming languages, the variables and constants declared in the main line logic are
   a. local to the main program
   b. unnamed
   c. never assigned a value
   d. global

3. Usually, the most difficult comparison operator to work with is ____.
   a. equal to
   b. greater than
   c. less than
   d. not equal to

4. All selection statements must have____.
   a. an if clause
   b. an else clause
   c. Both of a and b
   d. Non of above

5. When you use a range check, you compare a variable to the____ value in the range.
   a. lowest
   b. middle
   c. highest
   d. lowest or highest

6. Which of the following is NOT a decision making statement?
   a. if statement
   b. if-else statement
   c. for-loop
   d. switch statement

7. In any Boolean expression, the two values compared can be____.
   a. Variables
   b. Constants
   c. a and b
   d. none of above

8. Which one of followings is "less than or equal to" operator in C?
   a. =<
   b. <=
   c. ≤
   d. All of above

9. Which one of followings is "not equal to" operator in C?
   a. ≠
   b. =!
   c. !=

d. ==

10. Assuming that x=11, y=3, and z=7 initially, what will be their values after executing the following code segments?
```
if(x && y)
      x=10;
else
     y=10;
```
a. 11, 3, 7
b. 10, 10, 7
c. 11, 10, 7
d. None of above

11. Assuming that x=7, y=3, and z=5 initially, what will be their values after executing the following code segments?
```
if(x || y || z)
       y=10;
else
       z=0;
```
a. 7, 10, 0
b. 7, 3, 5
c. 7, 10, 5
d. None of the above

12. Assuming that x=1, y=0, and z=3 initially, what will be their values after executing the following code segments?
```
if(x)
     if(y)
       z=10;
else
     z=0;
```
a. 1, 0, 0
b. 1, 0, 10
c. 1, 0, 3
d. None of above

13. Assuming that x=11, y=3, and z=7 initially, what will be their values after executing the following code segments?
```
if(!y)
    z=0;
else
    y=1;
```
a. 11, 3, 0
b. 11, 1, 7
c. 11, 1, 0
d. 11, 3, 7

14. Assuming that x=5, y=3, and z=1 initially, what will be their values after executing the following code segment?
```
switch(x)
```

```
{
   case 5:
        x=1;
        y=x+1;
   case 3:
        x=0;
        break;
   case 1:
        z=y+x;
        break;
   default:
       x=1;
       y=0;
}
```

a. 5, 3, 1
b. 1, 3, 3
c. 1, 0, 1
d. None of the above

15. Assuming that x=5, y=3, and z=1 initially, what will be their values after executing the following code segment?

```
switch(x)
{
   case 1:
        x=0;
        y=0;
   case 5:
        x=2;
        z=2;
   default:
       x=1;
       y=2;
}
```

a. 2, 3, 2
b. 1, 2, 3
c. 1, 2, 0
d. None of above

16. Which of the following statements can be used to branch unconditionally from one point to another in the program?
a. if
b. if…else
c. for
d. goto

17. What will be the output of the following program?

```
main()
{
   int i;
   i=7;
   printf("%d", (i%2) ? i : i*2);
}
```
a. 14
b. 7
c. i
d. None of the above

18. What will be the output of the following program?

```
main()
{
   int i=1;
   if(i==1)
   {
      printf("B.C.A.  ");
      if(i==2)
          printf("M.C.A.  ");
      else
          printf("M.Sc.IT  ");
   }
   else
       printf("M.Sc.CA  ");
}
```
a. B.C.A. M.C.A. M.Sc.IT M.Sc.CA
b. B.C.A. M.C.A. M.Sc.CA
c. B.C.A. M.Sc.IT
d. B.C.A. M.Sc.IT M.Sc.CA
e. None of the above

Fill in the Blanks

1. A program in which one operation follows another from the beginning until the end is a ____ program.
2. The first type of documentation usually created when writing a program pertains____.
3. The selection structure is sometimes called ____.
4. Usually, the most difficult comparison operator to work with is____.
5. Symbols such as > and < are known as____.
6. All logical operators are binary operators except____ operator.
7. The ____ statement transfers the control out of the switch statement.
8. ____ operator is useful for making two-way decisions in C.
9. ____ statement is used to branch unconditionally from one point to another in the program.
10. Assuming x=1 initially, then after executing y=x++, y = ____ and x = ____.
12. Assuming x=1 initially, then after executing y=++x, y = ____ and x = ____.
13. The expression ! (x != y) can be replaced by the expression____.
14. ? : operator is popularly known as ____ operator.
15. Case labels must end with ____.

16. For a good programming practice, align vertically else clause with their matching ____ clause.
17. A temporary variable is also known as ____.
18. A Boolean expression is one that represents only one of two states, usually expressed as ____ or ____.
19. An expression like a > 7 is a(n) ____ expression.

True/False

1. Values can also be compared if they are of different type.
2. When you combine AND and OR operators, the OR operators take precedence.
3. The general rule for AND selection is: In an AND decision, first ask the question that is less likely to be true.
4. The general rule for OR selection is: In an OR decision, first ask the question that is more likely to be true.
5. One if can have one or more else clause.
6. The default case is required in the switch statement.
7. There can be more than one default labels in switch statement.
8. It is permitted to nest switch statements.
9. The predicate !((x>=10) | (y==5)) is equivalent to (x<10)&&(y!=5).
10. If there is only one statement to be executed depending on a condition, it must be enclosed within curly braces, {and}.
11. !(a= = b) is equal to (a!=b).
12. else is always paired with the most recent unpaired if.
13. The *label:* can be anywhere in the program either before or after the **goto** label; statement.
14. Case labels must be constants and cannot be constant expressions.
15. Case labels must be unique and so no two labels can have the same values.
16. && and & both are logical AND operators in C.
17. && is unary operator.
18. ! is a unary operator.
19. x += 1 is equal to x=x+1.
20. x *= 1 is equal to x=1;

UNIT : Looping and control breaks

Short Questions

1. What do you mean by loop?
2. Draw the flow chart for loop structure.
3. What do you mean by loop body?
4. Which variable is tested frequently in loop?
5. What are the advantages of loop?
6. List the loop available in C.
7. What do you mean by posttest loop?
8. What do you mean by pretest loop?
9. What are the different ways to control loop?
10. Define counter control loop.
11. What do you mean by sentinel value control loop?
12. What do you mean by definite loop?
13. Define indefinite loop.
14. What are the two common characteristics shared by both type of loop pretested as well as post tested?
15. List the common mistakes done in loop.
16. What are the common applications of loop structure?
17. Which loop is executed at least once without bothering the loop condition?
18. Give the example of the infinite loop.
19. Give the general structure of for loop.
20. Give the example of posttest and pretest loop.
21. What are the three steps found in every loop?
22. When can a structured loop be exited?
23. If the loop control variable is not initialized, what may happen?

Long Questions

1. Write a short note on loop.
2. What is the difference between posttest loop and pretest loop?
3. Explain While loop with proper example.
4. Explain do… while loop.
5. Explain for loop.
6. Explain nested loop.
7. What are the common mistakes done in loop structure?
8. Write a short note on application of loop.
9. Explain loop with respect to data validation.
10. Explain loop with respect to accumulation.
11. Write a program to print following pattern with given height(number of row) & width(number of characters):
```
+11111111+
*00000000*
*00000000*
*00000000*
+11111111+
```
12. Write a program to print numbers in reverse order. [N=5 then 5 4 3 2 1]
13. Write a program to check whether the number is prime number or not.
14. Write a program to reverse the given number.[N=123 then reverse number=321]
15. Write a program to print the following pattern:
```
N=5
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```
16. Write a program to generate following sequence:
    I. 6,10,14,18,22,26,…,50
    II. 2, 12, 36, 80, 150, …
17. Write a program to find the factorial.
18. Write a program to sum the first 15 odd numbers.
19. Write a menu driven program containing the operations: Addition, Subtraction, Multiplication, and Division.
20. Write a program to take 3 subject marks and then calculate overall result. How many percentage students are in distinction, first class, second class, pass class, fail.

Multiple Choice

1. The structure that allows you to write one set of instructions that operates on multiple, separate sets of data is the _____.
   a. sequence
   b. selection
   c. loop
   d. case

2. The loop that frequently appears in a program's mainline logic _____.
   a. always depends on whether a variable equals 0
   b. works correctly based on the same logic as other loops
   c. is an unstructured loop
   d. is an example of an infinite loop
3. Which of the following is not a step that must occur with every correctly-working loop?
   a. Initialize a loop control variable before the loop starts.
   b. Set the loop control value equal to a sentinel during each iteration.
   c. Compare the loop control value to a sentinel during each iteration.
   d. Alter the loop control variable during each iteration.
4. The statements executed within a loop are known collectively as the _____.
   a. loop body
   b. loop controls
   c. sequences
   d. sentinels
5. A counter keeps track of _____.
   a. the number of times an event has occurred
   b. the number of machine cycles required by a segment of a program
   c. the number of loop structures within a program
   d. the number of times software has been revised
6. Adding 1 to a variable is also called _____ it.
   a. digesting
   b. resetting
   c. decrementing
   d. incrementing
7. Which of the following is a definite loop?
   a. a loop that executes as long as a user continues to enter valid data
   b. a loop that executes 1,000 times
   c. both of the above
   d. none of the above
8. Which of the following is an indefinite loop?
   a. a loop that executes exactly 10 times

b. a loop that follows a prompt that asks a user how many repetitions to make and uses that value to control the loop

c. both of the above

d. none of the above

9. When you decrement a variable, you _____.

a. set it to 0

b. reduce it by one-tenth

c. subtract 1 from it

d. remove it from a program

10. When two loops are nested, the loop that is contained by the other is the _____ loop.

a. captive

b. unstructured

c. inner

d. outer

11. When loops are nested, _____.

a. they typically share a loop control variable

b. one must end before the other begins

c. both must be the same type—definite or indefinite

d. none of the above

12. Most programmers use a for loop _____.

a. for every loop they write

b. when a loop will not repeat

c. when they do not know the exact number of times a loop will repeat

d. when they know the exact number of times a loop will repeat

13. A report that lists no details about individual records, but totals only, is a(n) _____ report.

a. accumulator

b. final

c. summary

d. detailless

14. Typically, the value added to a counter variable is _____.

a. 0

b. 1

c. 10

d. 100

15. Typically, the value added to an accumulator variable is _____.

a. 0

b. 1

c. the same for each iteration

d. different in each iteration

16. After an accumulator or counter variable is displayed at the end of a program, it is best to _____.

a. delete the variable from the program

b. reset the variable to 0

c. subtract 1 from the variable

d. none of the above

17. When you _____, you make sure data items are the correct type and fall within the correct range.

a. validate data

b. employ offensive programming

c. use object orientation

d. count loop iterations

18. Overriding a user's entered value by setting it to a predetermined value is known as ______.

a. forcing

b. accumulating

c. validating

d. pushing

19. To ensure that a user's entry is the correct data type, frequently you ______.

a. prompt the user, asking if the user is sure the type is correct

b. use a method built into the programming language

c. include a statement at the beginning of the program that lists the data types allowed

d. all of the above

20. Variables might hold incorrect values even when they are _____.

a. the correct data type

b. within a required range

c. coded by the programmer rather than input by a user

d. all of the above

Fill in the blanks

1. Another term for “counting down” is ________.
2. If the loop control variable is not altered within the loop, _____ loop will occur.
3. In a for loop, the condition is tested ____[before or after] the loop body is executed?
4. Loops are frequently used to ____ and ____.
5. A report that contains only totals and other overall statistical information is called a(n) _____ report.
6. A variable that is used to hold a running total is called a(n) ____ variable.
7. A loop whose repetitions are managed by a counter is called ____ loop.
8. Any numeric variable you use to count the number of times an event has occurred is called ____.
9. One for which you cannot predetermine the number of executions is called ____ loop.
10. ____ loop is contained within another when loops are nested.
11. ____ loop contains the loop when loops are nested.
12. A variable that determines whether a loop will continue is called ____
13. ____ occur when a loop structure exists within another loop structure.
14. A number you use to increase a loop control variable on each pass through a loop is called ____.
15. In every loop, the value of ____ must be altered.
16. ____ is posttest loop.
17. In ____ loop, iteration is counted.
18. A variable is decreasing it by a constant value, frequently 1 is called ____.
19. Another term for “counting up” is _____.
20. ____ loop contain initialization, condition, and increment contained like a package.

True /False

1. Nested loop is not a structure.
2. Incrementing or decrementing of variable in event control loop is necessary.
3. Event control loop is always indefinite loop.
4. For loop is posttest loop.
5. In while loop condition is checked first.
6. Loop body is executed once without checking the condition in for loop.
7. Do while loop is pretest loop.
8. Loop control variable must be altered in a loop structure.
9. Initialization of loop control variable is must in any finite loop.
10. The best loop for data validation is do…while loop.
11. In pretest loop, processing is done one or more time.
12. Comma expression is not allowed in for loop initialization.
13. If an event must occur to terminate a loop then it is called counter control loop.
14. Minimum iteration in for loop is one.
15. In pretest counter control loop, loop control variable is tested n+1 times.
16. Do …until loop exists in C.
17. Data validation is an application of loop.
18. while (true); is finite event control loop.
19. while(false); is definite loop.
20. Inner loop is the first loop in the program.

UNIT : Arrays

Short Question

1. What do you mean by array?
2. What do you mean by a subscript with respect to array?
3. Define string. Give the example of it with memory allocation.
4. How can we access the element of the array?
5. Define parallel array.
6. What do you mean by element?
7. How can we access each element of the array?
8. How many elements are there in a[5][7]?
9. What is the range of the a[10]?
10. When will you say that subscript is out of the bound?
11. What is the length of the null string?
12. Give formula to calculate the address of the single dimension array.
13. Is it necessary to declare the size of the array at compile time?

14. How can you read a line of text containing a variety of characters, including white spaces, using scanf? Explain in brief.
15. Differentiate strcpy and strncpy functions.
16. Differentiate strcmp and strncmp functions.
17. Differentiate strcat and strncat functions.

Long Question

1. Write a short note on array.
2. Explain with example array declaration & initialization of the array.
3. What do you mean by string? Write a code to find the length of the string.
4. Write a code to print each element of the array.
5. Write an algorithm to sort elements of an array in ascending order.
6. Write a code to reverse the string.
7. Explain the constant array with appropriate example.
8. Explain parallel array with example.
9. Write a note on searching for the range match.
10. Which are the different ways to pass the array to the function?
11. Give the example of the C code explaining the static variable concept.
12. Write a C code to copy the string into another string.
13. Explain any five functions of string.h with example.
14. Explain any five functions of math.h with example.
15. Explain multi dimensional array with appropriate example.
16. What is the difference between integer array & character array?
17. Explain two ways of initialization of one-dimensional array.
18. Describe different ways to initialize two-dimensional arrays.
19. Describe the limitations of using getchar and scanf functions for reading strings.
20. Explain array of characters.
21. Explain array of strings.

Multiple Choice

1. A subscript is a(n) _____.
   a. element in an array
   b. alternate name for an array
   c. number that represents the highest value stored within an array
   d. number that indicates the position of a particular item in an array

2. Each variable in an array must have the same _____ as the others.

a. data type
b. subscript
c. value
d. memory location

3. Each data item in an array is called a(n) _____.

a. data type
b. subscript
c. component
d. element

4. The subscripts of any array are always _____.

a. integers
b. fractions
c. characters
d. strings of characters

5. Suppose you have an array named number, and two of its elements are number[1] and number[4]. You know that _____.

a. the two elements hold the same value
b. the array holds exactly four elements
c. there are exactly two elements between those two elements
d. the two elements are at the same memory location

6. The most useful type of subscript for manipulating arrays is a _____.

a. numeric constant
b. variable
c. character
d. filename

7. A program contains a seven-element array that holds the names of the days of the week. At the start of the program, you display the day names using a subscript named dayNum. You display the same array values again at the end of the program, where you _____ as a subscript to the array.

a. must use dayNum
b. can use dayNum, but can also use another variable
c. must not use dayNum
d. must use a numeric constant instead of a variable

8. Filling an array with values during a program's execution is known as _____ the array.
   a. executing
   b. colonizing
   c. populating
   d. declaring
9. Using an array can make a program _____.
   a. easier to understand
   b. illegal in some modern languages
   c. harder to maintain
   d. all of the above
10. What do you call two arrays in which each element in one array is associated with the element in the same relative position in the other array?
    a. cohesive arrays
    b. parallel arrays
    c. hidden arrays
    d. perpendicular arrays
11. In most modern programming languages, the highest subscript you should use with a 10-element array is _____.
    a. 8
    b. 9
    c. 10
    d. 11
12. Parallel arrays _____.
    a. frequently have an indirect relationship
    b. never have an indirect relationship
    c. must be the same data type
    d. must not be the same data type
13. Each element in a five-element array can hold _____ value(s).
    a. one
    b. five
    c. at least five
    d. an unlimited number of

14. When you use a subscript value that is negative or higher than the number of elements in an array, _____.
    a. execution of the program stops and an error message is issued
    b. a value in a memory location that is outside the area occupied by the array will be accessed
    c. a value in a memory location that is outside the area occupied by the array will be accessed, but only if the value is the correct data type
    d. the resulting action depends on the programming language used

15. In every array, a subscript is out of bounds when it is _____.
    a. negative
    b. 0
    c. 1
    d. 999

16. You can access every element of an array using a _____.
    a. while loop
    b. for loop
    c. both of the above
    d. none of the above

17. _____ is used to get length of string.
    a. strstr
    b. strcat
    c. strlen
    d. strcmp

18. _____ is used to assign a string to another string.
    a. strstr
    b. strcmp
    c. strcat
    d. strcpy

19. _____ is used for string comparison.
    e. strstr
    f. strcmp

g. strcat

h. strcpy

20. Which of the following can be returned by strcmp function?

    a. Positive integer
    b. Negative integer
    c. Zero
    d. All of the above: a, b, and c
    e. None of the above

Fill in the blanks

1. Each element of an array has the same ______ and the same ______.
2. Each element of an array has a unique ______.
3. ______ indicates how far away an element is from the first element.
4. A _____ is a variable that you set to indicate whether some event has occurred.
5. ______ is a programming module that has the series of the statement to perform a task.
6. ______ is the process of arranging the elements of an array in order.
7. The variable used as a subscript in an array is popularly known as ______ variable.
8. A ______ is a sequence of characters that is treated as a single data item.
9. A string is represented as array of ______.
10. A string terminates with ______character.
11. We can use the conversion specification ______ in scanf to read a line of text.
12. The function ______ is used to determine the length of a string.
13. The printf can be replaced by ______ function for printing strings.
14. ASCII stands for ______.
15. A null character ('\0') has the ASCII value ______.
16. ______ terminates the string.
17. C can calculate the address of any element in the array using the formula ______.
18. Array where the data are organized linearly in only one direction is known as ______ array.
19. We need to specify three things, namely, ______ , ______ , and ______ , when we declare an array.

True/False

1. Array is the derived data type.
2. An array is a sequence of memory location of same data type.
3. String is terminated by the null character '\0'.
4. The type of all elements in an array must be the same type.
5. When an array is declared, C automatically initializes its elements to zero.
6. Array is the collection of the elements having different data type.
7. Index of the array starts from 1.
8. Array elements are not stored in a sequential manner.
9. Array can be initialized at compile time as well as run time.
10. If we try to store the value out of the index then it will give error at compile time.
11. Null value is the last value in any type of the array.
12. When the integer array is the partially initialized the rest of the elements are initialize with zero.
13. C checks the validity of the subscript when you use array.
14. Strings are array of characters.
15. Subscripts begin at 0 and end at -1.
16. A char type variable cannot be used as a subscript in an array.

UNIT: Methods

Short Questions

1. What do you mean by modularization?
2. List at least three advantages of modularization.
3. Define Abstraction
4. List at least two rules for naming modules.
5. Which are the components of a method that a method must include?
6. Describe three distinct parts of mainline logic.
7. Name the tool used to show how modules are related to one another.
8. List two different ways to pass array to the function.
9. Define the following terms:
   a. Method
   b. Argument
   c. Parameter
10. Return type of the function cannot be a pointer. True/false? Justify.
11. What do you mean by a recursive function?
12. Which two items must be included within the method declaration parentheses?
13. List the three elements of a function in C.
14. List elements of a function definition in C.
15. What is local variable in terms of functions?
16. Differentiate formal parameters and actual parameters.

Long Questions

1. Explain advantages of modularization.
2. Explain the use of method with an example.
3. Write a short note on hierarchy chart.
4. Explain function having multiple parameters with example.
5. Explain function with return type.
6. Explain how to pass an array to a method.
7. Explain pass by value and pass by reference, ways to call a method.
8. Discuss formal parameters and actual parameters.
9. Write a program to accept values for array elements from the user and create method that reverses the order of values in array. Display reversed array in main program.
10. Explain recursive function with suitable example.
11. Explain elements of function definition with example.
12. Explain function declaration.
13. Write a program to create a function that works as pow() function of math.h

Multiple Choice Questions

1. Which of the following is not a term used as a synonym for “module” in any programming language?
   a. Structure
   b. Procedure
   c. Method
   d. Function

2. Which of the following is not a reason to use modularization?
   a. Modularization provides abstraction.
   b. Modularization allows multiple programmers to work on a problem.
   c. Modularization allows you to reuse your work.
   d. Modularization eliminates the need for structure.

3. Which of the following is true?
   a. A program can call, at most, one method.
   b. A program can contain a method that calls another method.
   c. A method can contain one or more other methods.
   d. All of these are true.

4. Which of the following must every method have?
   a. a header
   b. a parameter list
   c. a return value
   d. All of these

5. Which of the following is most closely related to the concept of "local"?
   a. Abstract
   b. object-oriented
   c. in scope
   d. program level

6. Although the terms parameter and arguments are closely related, the difference between them is that "argument" refers to _____.
   a. a passed constant
   b. a value in a method call
   c. a formal parameter
   d. a variable that is local to a method
7. A method's interface is its _____.
   a. signature
   b. return type
   c. identifier
   d. parameter list
8. When you write the method declaration for a method that can receive a parameter, which of the following must be included in the method declaration?
   a. the name of the argument that will be used to call the method
   b. a local name for the parameter
   c. the return value for the method
   d. All of these
9. When you use a variable name in a method call, it _____ the same name as the variable in the method header.
   a. can have
   b. cannot have
   c. must have
   d. must not have
10. Assume you have written a method with the header void myMethod(num a, string b). Which of the following is a correct method call?
    a. myMethod(12)
    b. myMethod(12, "Hello")
    c. myMethod("Goodbye")
    d. It is impossible to tell.
11. Assume you have written a method with the header num yourMethod(string name, num code). The

method’s type is _____.

a. num

b. string

c. num and string

d. void

12. Assume you have written a method with the header string myMethod(num score, string grade). Also assume you have declared a numeric variable named test. Which of the following is a correct method call?

a. myMethod()

b. myMethod(test)

c. myMethod(test, test)

d. myMethod(test,"A")

13. The value used in a method's return statement must _____.

a. be numeric

b. be a variable

c. match the data type used before the method name in the header

d. Two of the above

14. When a method receives a copy of the value stored in an argument used in the method call, it means the variable was _____.

a. unnamed

b. passed by value

c. passed by reference

d. assigned its original value when it was declared

15. A void method _____.

a. contains no statements

b. requires no parameters

c. returns nothing

d. has no name

16. When an array is passed to a method, it is _____.

a. passed by reference

b. passed by value
c. unnamed in the method
d. unalterable in the method

Fill in the blanks

1. In most modern programming languages, when a variable or a constant declared in a method, the variable or constant is ____ in that method.
2. A method's ____ is known more succinctly as a method's type.
3. A pointer variable contains as its value the ______ of another variable.
4. The ______ operator returns the value of the variable to which its operand points.
5. The ______ operator is used with a pointer to de-reference the address contained in the pointer.
6. A ______ is a self-contained block of code that performs a particular task.
7. The ______ and ______ arguments should match in number, type, and order.
8. The variables declared inside a function are known as ______ variables.
9. C function returns a value of the type ______ as the default case when no other type is specified explicitly.
10. The operator * is known as ______ operator.
11. A function can be called by using the function name followed by a list of ______ parameters.
12. When a called function in turn calls another function a process of 'chaining' occurs, which is known as ______.
13. Another name for the indirection operator is ______ operator.

True/False

1. Array cannot be passed to the function because its structure is too complicated.
2. Only the size of the first dimension is needed when a two dimensional array is declared in a parameter list.
3. When the array is passed to the function it is always passed pass by reference.
4. There is no difference between argument and parameter.
5. Method can return more than one value.
6. Argument and parameters must have the same order in a method.
7. Entire array can be passed by value to a method.
8. A program can call, at most, one method.
9. When a method returns a value, the method must have a return type.
10. A method's declared return type must match the type of the value used in the return statement.
11. You can use a method's return value directly, without storing it.
12. A method can require at most one parameter.
13. The types of the actual and formal arguments must be same.

UNIT: Derived Types and File Handling

Short Questions

1. What do you mean by pointer? Explain with proper example for declare as well as initialize the pointer variable.
2. Write a code containing function to swap the value of two variable & print its value in

main().

3. Give example of the function which returns the pointer.
4. In which situation you will prefer pointer. Explain with proper example.
5. Explain array of pointer.
6. What do you mean by pointer?
7. How many bytes are allocated to pointer?
8. What are the needs of the pointer?
9. Differentiate getc and getw functions in C.
10. Differentiate putc and putw functions in C.
11. Describe the use of fprintf and fscanf functions.
12. How to find the size of a structure?
13. What do you mean by preprocessors?

Long Questions

1. Write a code to swap two values using pointer.
2. Explain enumerated (enum) data type in C.
3. Discuss with example how to declare and initialize structure.
4. Explain how to access members of a structure.
5. Differentiate array of structure and array within the structure.
6. Compare structure and union in C with example.
7. Explain functions used to open and close file with example.
8. Discuss input/output operations on file.
9. Compare getc, getw, and fscanf.
10. Compare putc, putw, and fprintf.
11. Explain fssek, ftell, and rewind functions used in random access to file.
12. Explain command line arguments.
13. Write a program to read the content of one file and write it in another file.
14. Explain macro substitution.
15. Explain file inclusion directives.
16. Explain compiler control directives.

Multiple Choice Questions

1. Which of the following code segments correctly swaps the values of variables named x and y?

a. x = y
   y = temp
   x = temp

b. temp = x
   x = y
   y = temp

c. x = y
   temp = x
   y = temp
d. temp = x
   y = x
   x = temp

2. Which of the following operator is known as address of operator?
   a. &
   b. *
   c. ->
   d. **
3. Which of the following operator known as indirection operator?
   a. &
   b. <<
   c. !
   d. *
4. Identify the correct declarations of pointer variables ptr1, ptr2.
   a. int ptr1, *ptr2;
   b. int *ptr1, ptr2;
   c. int ptr1, ptr2;
   d. int *ptr1, *ptr2;

5. Pointer variables may be assigned
   a. address value represented in hexadecimal notation.
   b. address value represented in octal notation
   c. address of another variable
   d. address value represented in binary notation

6. Operand of indirection operator is
   a. pointer variable
   b. ordinary variable
   c. integer variable
   d. none of above

7. Pointer can be used to achieve

a. call by function
b. call by value
c. call by reference
d. none of above

8. int *p, a[10]; p = a; Which of the following is incorrect way to access the 5th element of the array?
   a. *(a+4)
   b. a[4]
   c. *(*p+4)
   d. p[4]

9. The statement "int * p ;" can be interpreted as
   a. the variable whose address is stored in p is an integer.
   b. the variable pointed to by p is an integer.
   c. p points to an integer.
   d. All of above.

10. For the statement "p = *m;" which of the following statements is TRUE?
    a. The value of m is assigned to p.
    b. The value of the variable pointed to by m is assigned to p.
    c. p is a pointer variable.
    d. Address of m is assigned to p.

11. Identy the syntax error in the following code segment.
    int *ptr, m = 100; printf("%d",*ptr);
    a. Declaring ptr and m in one line is wrong
    b. In line 2, *ptr should be written as &ptr
    c. The address of m should be assigned to ptr before it is accessed
    d. No error

12. Which of the following is the correct way of declaring a float pointer?
    a. float ptr;
    b. float *ptr;
    c. *float ptr;
    d. None if the above

13. To open a file, _______ function is used.
    a. fclose
    b. fopen
    c. fileopen
    d. All of the above

Fill in the blanks

1. The mode ____ is used for opening a file for updating.
2. The function ____ may be used to position a file at beginning.
3. The function ____ gives the current position in the file.
4. The function ____ is used to write data to randomly accessed file.
5. The _______ directive discords a macro.
6. The operator _______ is used to concatenate two arguments.
7. The operator _______ converts its operand.
8. The _______ directive causes an implementation-oriented action.

True/False

1. FILE is a defined data type.
2. Files are always referred to by name in C programs.
3. Using fseek to position a file beyond the end of the file is an error.
4. Function fseek may be used to seek from beginning of the file only.
5. Pointer constants are the addresses of memory locations.
6. Pointer variables are declared using the address operator.
7. The underlying type of a pointer variable is void.
8. A file must be open before it can be used.
9. All files must be explicitly closed.
10. The keyword #define must be written starting from the first column.
11. Like other statements, a processor directive must end with a semicolon.
12. All preprocessor directives begin with #.
13. We cannot use a macro in the definition of another macro.