**CITC-1301 Introduction to Programming**

**Midterm Review**

## Chapter 1 – Introduction to Computers and Programming

1. A(n) \_\_\_\_\_\_\_\_\_\_ is a set of instructions that a computer follows to perform a task.
   1. compiler
   2. program
   3. interpreter
   4. programming language
2. The physical devices that a computer is made of are referred to as \_\_\_\_\_\_\_\_\_\_.
   1. hardware
   2. software
   3. the operating system
   4. tools
3. The part of a computer that runs programs is called \_\_\_\_\_\_\_\_\_\_.
   1. RAM
   2. secondary storage
   3. main memory
   4. the CPU
4. Today, CPUs are small chips known as \_\_\_\_\_\_\_\_\_\_.
   1. ENIACs
   2. microprocessors
   3. memory chips
   4. operating systems
5. The computer stores a program while the program is running, as well as the data that the program is working with, in \_\_\_\_\_\_\_\_\_\_.
   1. secondary storage
   2. the CPU
   3. main memory
   4. the microprocessor
6. This is a volatile type of memory that is used only for temporary storage while a program is running.
   1. RAM
   2. secondary storage
   3. the disk drive
   4. the USB drive
7. A type of memory that can hold data for long periods of time, even when there is no power to the computer, is called \_\_\_\_\_\_\_\_\_\_.
   1. RAM
   2. main memory
   3. secondary storage
   4. CPU storage
8. A component that collects data from people or other devices and sends it to the computer is called \_\_\_\_\_\_\_\_\_\_.
   1. an output device
   2. an input device
   3. a secondary storage device
   4. main memory
9. A video display is a(n) \_\_\_\_\_\_\_\_\_\_ device.
   1. output
   2. input
   3. secondary storage
   4. main memory
10. A \_\_\_\_\_\_\_\_\_\_ is enough memory to store a letter of the alphabet or a small number.
    1. byte
    2. bit
    3. switch
    4. transistor
11. A byte is made up of eight \_\_\_\_\_\_\_\_\_\_.
    1. CPUs
    2. instructions
    3. variables
    4. bits
12. In the \_\_\_\_\_\_\_\_\_\_ numbering system, all numeric values are written as sequences of 0s and 1s.
    1. hexadecimal
    2. binary
    3. octal
    4. decimal
13. A bit that is turned off represents the following value: \_\_\_\_\_\_\_\_\_\_.
    1. 1
    2. −1
    3. 0
    4. “no”
14. A set of 128 numeric codes that represent the English letters, various punctuation marks, and other characters is \_\_\_\_\_\_\_\_\_\_.
    1. binary numbering
    2. ASCII
    3. Unicode
    4. ENIAC
15. An extensive encoding scheme that can represent characters for many languages in the world is \_\_\_\_\_\_\_\_\_\_.
    1. binary numbering
    2. ASCII
    3. Unicode
    4. ENIAC
16. The tiny dots of color that digital images are composed of are called \_\_\_\_\_\_\_\_\_\_.
    1. bits
    2. bytes
    3. color packets
    4. pixels
17. If you were to look at a machine language program, you would see \_\_\_\_\_\_\_\_\_\_.
    1. Python code
    2. a stream of binary numbers
    3. English words
    4. circuits
18. Computers can only execute programs that are written in \_\_\_\_\_\_\_\_\_\_.
    1. Java
    2. assembly language
    3. machine language
    4. Python
19. The \_\_\_\_\_\_\_\_\_\_ translates an assembly language program to a machine language program.
    1. assembler
    2. compiler
    3. translator
    4. interpreter
20. The words that make up a high-level programming language are called \_\_\_\_\_\_\_\_\_\_.
    1. binary instructions
    2. mnemonics
    3. commands
    4. key words
21. The rules that must be followed when writing a program are called \_\_\_\_\_\_\_\_\_\_.
    1. syntax
    2. punctuation
    3. key words
    4. operators
22. A(n) \_\_\_\_\_\_\_\_\_\_ program translates a high-level language program into a separate machine language program.
    1. assembler
    2. compiler
    3. translator
    4. utility
23. (True|False) Today, CPUs are huge devices made of electrical and mechanical components such as vacuum tubes and switches.
24. (True|False) Main memory is also known as RAM.
25. (True|False) Any piece of data that is stored in a computer’s memory must be stored as a binary number.
26. (True|False) Images, like the ones created with your digital camera, cannot be stored as binary numbers.
27. (True|False) Machine language is the only language that a CPU understands.
28. (True|False) Assembly language is considered a high-level language.
29. (True|False) An interpreter is a program that both translates and executes the instructions in a high-level language program.
30. (True|False) A syntax error does not prevent a program from being compiled and execute
31. (True|False) Windows, Linux, Android, iOS, and macOS are all examples of application software.
32. (True|False) Word processing programs, spreadsheet programs, email programs, web browsers, and games are all examples of utility programs.
33. How does the main memory and the secondary storage of a computer differ from each other?
34. What number does a bit that is turned on represent? What number does a bit that is turned off represent?
35. How many different characters can be represented in ASCII? Name the character set that addresses this limitation.
36. What is an individual instruction in a program written in a high-level programming language called?
37. What are the short words that are used in assembly language called?
38. What is the difference between a compiler and an interpreter? Compiler turns it into machine code at once, before program runs. Interpreter converts during program run.
39. What type of software controls the internal operations of the computer’s hardware? – operating system

## Chapter 2 – Input, Processing, and Output

1. A \_\_\_\_\_\_\_\_\_\_ error does not prevent the program from running but causes it to produce incorrect results.
   1. syntax
   2. hardware
   3. logic
   4. fatal
2. A \_\_\_\_\_\_\_\_\_\_ is a single function that the program must perform in order to satisfy the customer.
   1. task
   2. software requirement
   3. prerequisite
   4. predicate
3. A(n) \_\_\_\_\_\_\_\_\_\_ is a set of well-defined logical steps that must be taken to perform a task.
   1. logarithm
   2. plan of action
   3. logic schedule
   4. algorithm
4. An informal language that has no syntax rules and is not meant to be compiled or executed is called \_\_\_\_\_\_\_\_\_\_.
   1. faux code
   2. pseudocode
   3. Python
   4. a flowchart
5. A \_\_\_\_\_\_\_\_\_\_ is a diagram that graphically depicts the steps that take place in a program.
   1. flowchart
   2. step chart
   3. code graph
   4. program graph
6. A \_\_\_\_\_\_\_\_\_\_ is a sequence of characters.
   1. char sequence
   2. character collection
   3. string
   4. text block
7. A \_\_\_\_\_\_\_\_\_\_ is a name that references a value in the computer’s memory.
   1. variable
   2. register
   3. RAM slot
   4. byte
8. A \_\_\_\_\_\_\_\_\_\_ is any hypothetical person using a program and providing input for it.
   1. designer
   2. user
   3. guinea pig
   4. test subject
9. A string literal in Python must be enclosed in \_\_\_\_\_\_\_\_\_\_.
   1. parentheses.
   2. single-quotes.
   3. double-quotes.
   4. either single-quotes or double-quotes.
10. Short notes placed in different parts of a program explaining how those parts of the program work are called \_\_\_\_\_\_\_\_\_\_.
    1. comments
    2. reference manuals
    3. tutorials
    4. external documentation
11. A(n) \_\_\_\_\_\_\_\_\_\_ makes a variable reference a value in the computer’s memory.
    1. variable declaration
    2. assignment statement
    3. math expression
    4. string literal
12. This symbol marks the beginning of a comment in Python.
    1. &
    2. \*
    3. \*\*
    4. #
13. Which of the following statements will cause an error? – THE EQUALS SIGN IS CALLED THE ASSIGNMENT OPERATOR
    1. x = 17
    2. 17 = x
    3. x = 99999
    4. x = '17'
14. In the expression 12 + 7, the values on the right and left of the + symbol are called \_\_\_\_\_\_\_\_\_\_.
    1. operands
    2. operators
    3. arguments
    4. math expressions
15. This operator performs integer division.
    1. //
    2. % - modulo -returns remainder
    3. \*\* - to the power
    4. / - regular division
16. This is an operator that raises a number to a power.
    1. %
    2. \*
    3. \*\*
    4. /
17. This operator performs division, but instead of returning the quotient it returns the remainder.
    1. %
    2. \*
    3. \*\*
    4. /
18. Suppose the following statement is in a program: price = 99.0. After this statement executes, the price variable will reference a value of which data type?
    1. int
    2. float
    3. currency
    4. str
19. Which built-in function can be used to read input that has been typed on the keyboard?
    1. input()
    2. get\_input()
    3. read\_input()
    4. keyboard()
20. Which built-in function can be used to convert an int value to a float?
    1. int\_to\_float()
    2. float()
    3. convert()
    4. int()
21. A magic number is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
    1. a number that is mathematically undefined
    2. an unexplained value that appears in a program’s code
    3. a number that cannot be divided by 1
    4. a number that causes computers to crash
22. A \_\_\_\_\_\_\_\_\_\_ is a name that represents a value that does not change during the program’s execution.
    1. named literal
    2. named constant
    3. variable signature
    4. key term
23. (True|False) Programmers must be careful not to make syntax errors when writing pseudocode programs.
24. (True|False) In a math expression, multiplication and division take place before addition and subtraction.
25. (True|False) Variable names can have spaces in them.
26. (True|False) In Python, the first character of a variable name cannot be a number.
27. (True|False) If you print a variable that has not been assigned a value, the number 0 will be displayed.
28. What does a professional programmer usually do first to gain an understanding of a problem? - Stop and think/ ask the person requesting the program about the requirements.
29. How does pseudocode differ from actual code written in a programming language? - No syntax, not actual code, cannot be actually run.
30. Computer programs typically perform what three steps? – input, process, output
31. What rules and considerations should influence the names given to variables in a program? – make it meaningful and relevant to the use
32. What is the difference between floating-point division and integer division? – Float returns string. Integer returns integer
33. What is a magic number? Why are magic numbers problematic? – unexplained numeric literal appearing in code. hard to determine meaning, could be in multiple locations, if changed may need changed in multiple locations. Just create a constant.
34. Assume a program uses the named constant PI to represent the value 3.14159. The program uses the named constant in several statements. What is the advantage of using the named constant instead of the actual value 3.14159 in each statement? - easier to change. Refer to why magic numbers are bad.

## Chapter 3 – Decision Structures and Boolean Logic

1. A \_\_\_\_\_\_\_\_\_\_ structure can execute a set of statements only under certain circumstances.
   1. sequence
   2. circumstantial
   3. decision
   4. Boolean
2. A \_\_\_\_\_\_\_\_\_\_ structure provides one alternative path of execution.
   1. sequence
   2. single alternative decision
   3. one path alternative
   4. single execution decision
3. A(n) \_\_\_\_\_\_\_\_\_\_ expression has a value of either True or False.
   1. binary
   2. decision
   3. unconditional
   4. Boolean
4. The symbols >, <, and == are all \_\_\_\_\_\_\_\_\_\_ operators.
   1. relational
   2. logical
   3. conditional
   4. ternary
5. A(n) \_\_\_\_\_\_\_\_\_ structure tests a condition and then takes one path if the condition is true, or another path if the condition is false.
   1. if statement
   2. single alternative decision
   3. dual alternative decision
   4. sequence
6. You use a(n) \_\_\_\_\_\_\_\_\_\_ statement to write a single alternative decision structure.
   1. test-jump
   2. if
   3. if-else
   4. if-call
7. You use a(n) \_\_\_\_\_\_\_\_\_\_ statement to write a dual alternative decision structure.
   1. test-jump
   2. if
   3. if-else
   4. if-call
8. and, or, and not are \_\_\_\_\_\_\_\_\_\_ operators.
   1. relational
   2. logical
   3. conditional
   4. ternary
9. A compound Boolean expression created with the \_\_\_\_\_\_\_\_\_\_ operator is true only if both of its subexpressions are true.
   1. and
   2. or
   3. not
   4. both
10. A compound Boolean expression created with the \_\_\_\_\_\_\_\_\_ operator is true if either of its subexpressions is true.
    1. and
    2. or
    3. not
    4. either
11. The \_\_\_\_\_\_\_\_\_\_\_ operator takes a Boolean expression as its operand and reverses its logical value.
    1. and
    2. or
    3. not
    4. either
12. A \_\_\_\_\_\_\_\_\_\_\_ is a Boolean variable that signals when some condition exists in the program.
    1. flag
    2. signal
    3. sentinel
    4. siren
13. (True|False) You can write any program using only sequence structures.
14. (True|False) A program can be made of only one type of control structure. You cannot combine structures.
15. (True|False) A single alternative decision structure tests a condition and then takes one path if the condition is true, or another path if the condition is false.
16. (True|False) A decision structure can be nested inside another decision structure.
17. (True|False) A compound Boolean expression created with the and operator is true only when both subexpressions are true.
18. Explain what is meant by the term “conditionally execute”. – It only happens if some condition exists
19. Explain how a single alternative decision structure and a dual alternative decision structure differ. – Single has only one path. A single if. A dual has two possibilities. If/else.
20. Briefly describe how the and operator works. – If all subexpressions are true, the compound expression is true.
21. Briefly describe how the or operator works. – Only one of the subexpressions must be true for the compound expression to be true.
22. When determining whether a number is outside a range, which logical operator is it best to use? Logical or
23. What is a flag and how does it work? – a flag denotes when something is used in your program.
24. Write nested decision structures that perform the following: If amount1 is greater than 10 and amount2 is less than 100, display the greater of amount1 and amount2.
25. Write an if-else statement that assigns True to the again variable if the score variable is within the range of 40 to 49. If the score variable’s value is outside this range, assign False to the again variable.

## Chapter 4 – Repetition Structures

1. A \_\_\_\_\_\_\_\_\_\_ -controlled loop uses a true/false condition to control the number of times that it repeats.
   1. Boolean
   2. condition
   3. decision
   4. count
2. A \_\_\_\_\_\_\_\_\_\_ -controlled loop repeats a specific number of times.
   1. Boolean
   2. condition
   3. decision
   4. count
3. Each repetition of a loop is known as a(n) \_\_\_\_\_\_\_\_\_\_.
   1. cycle
   2. revolution
   3. orbit
   4. iteration
4. The while loop is a \_\_\_\_\_\_\_\_\_\_ type of loop.
   1. pretest
   2. no-test
   3. prequalified
   4. post-iterative
5. A(n) \_\_\_\_\_\_\_\_\_\_ loop has no way of ending and repeats until the program is interrupted.
   1. indeterminate
   2. interminable
   3. infinite – ctrl + C to interrupt
   4. timeless
6. The -= operator is an example of a(n) \_\_\_\_\_\_\_\_\_\_ operator.
   1. relational
   2. augmented assignment
   3. complex assignment
   4. reverse assignment
7. A(n) \_\_\_\_\_\_\_\_\_\_ variable keeps a running total.
   1. sentinel
   2. sum
   3. total
   4. accumulator
8. A(n) \_\_\_\_\_\_\_\_\_\_ is a special value that signals when there are no more items from a list of items to be processed. This value cannot be mistaken as an item from the list.
   1. sentinel
   2. flag
   3. signal
   4. accumulator
9. GIGO stands for \_\_\_\_\_\_\_\_\_\_.
   1. great input, great output
   2. garbage in, garbage out
   3. GIGahertz Output
   4. GIGabyte Operation
10. The integrity of a program’s output is only as good as the integrity of the program’s \_\_\_\_\_\_\_\_\_\_.
    1. compiler
    2. programming language
    3. input
    4. debugger
11. The input operation that appears just before a validation loop is known as the \_\_\_\_\_\_\_\_\_\_.
    1. prevalidation read
    2. primordial read
    3. initialization read
    4. priming read
12. Validation loops are also known as \_\_\_\_\_\_\_\_\_\_.
    1. error traps
    2. doomsday loops
    3. error avoidance loops
    4. defensive loops
13. (True|False) A condition-controlled loop always repeats a specific number of times.
14. (True|False) The while loop is a pretest loop. (so is for)
15. (True|False) The following statement subtracts 1 from x: x = x − 1
16. (True|False) It is not necessary to initialize accumulator variables.
17. (True|False) In a nested loop, the inner loop goes through all of its iterations for every single iteration of the outer loop.
18. (True|False) To calculate the total number of iterations of a nested loop, add the number of iterations of all the loops.
19. (True|False) The process of input validation works as follows: when the user of a program enters invalid data, the program should ask the user “Are you sure you meant to enter that?” If the user answers “yes,” the program should accept the data.
20. What is a condition-controlled loop? – While loop that executes a conditionally determined number of times until false.
21. What is a count-controlled loop? – will iterate a specified number of times.
22. What is an infinite loop? Write the code for an infinite loop. – a loop that executes forever. Generally a while loop.
23. How are an accumulator variable and a loop used to calculate a running total?
24. How many iterations would occur in total if a loop that repeats 5 times is nested within a loop that repeats 4 times? 20
25. Why must the value chosen for use as a sentinel be carefully selected? Has to be an out of bounds value that would never be entered. It tells the program to stop. Like a negative number when inputting grades.
26. What does the phrase “garbage in, garbage out” mean? – a bad input will produce a bad output.
27. Give a general description of the input validation process. – ask user for input. loops until input value is good. If bad, tells user how to fix their input and continues to ask.
28. Write a for loop that uses the range function to display all odd numbers between 1 and 100. For value in range(1, 101): if value %2 !=0 print(value)
29. Write code that prompts the user to enter a positive nonzero number and validates the input.

## Chapter 5 – Functions

1. A group of statements that exist within a program for the purpose of performing a specific task is a(n) \_\_\_\_\_\_\_\_\_\_.
   1. block
   2. parameter
   3. function
   4. expression
2. A design technique that helps to reduce the duplication of code within a program and is a benefit of using functions is \_\_\_\_\_\_\_\_\_\_.
   1. code reuse
   2. divide and conquer
   3. debugging
   4. facilitation of teamwork
3. The first line of a function definition is known as the \_\_\_\_\_\_\_\_\_\_.
   1. body
   2. introduction
   3. initialization
   4. header
4. You \_\_\_\_\_\_\_\_\_\_ a function to execute it.
   1. Define – creating it
   2. call
   3. import
   4. export
5. A design technique that programmers use to break down an algorithm into functions is known as \_\_\_\_\_\_\_\_\_\_.
   1. top-down design
   2. code simplification
   3. code refactoring
   4. hierarchical subtasking
6. A \_\_\_\_\_\_\_\_\_\_ is a diagram that gives a visual representation of the relationships between functions in a program.
   1. flowchart
   2. function relationship chart
   3. symbol chart
   4. hierarchy chart
7. A \_\_\_\_\_\_\_\_\_\_ is a variable that is created inside a function.
   1. global variable
   2. local variable
   3. hidden variable
   4. none of the above; you cannot create a variable inside a function
8. A(n) \_\_\_\_\_\_\_\_\_\_ is the part of a program in which a variable may be accessed.
   1. declaration space
   2. area of visibility
   3. scope
   4. mode
9. A(n) \_\_\_\_\_\_\_\_\_\_ is a piece of data that is sent into a function. ON THE EXAM
   1. argument
   2. parameter
   3. header
   4. packet
10. A(n) \_\_\_\_\_\_\_\_\_\_ is a special variable that receives a piece of data when a function is called.
    1. argument
    2. parameter
    3. header
    4. packet
11. A variable that is visible to every function in a program file is a \_\_\_\_\_\_\_\_\_\_.
    1. local variable
    2. universal variable
    3. program-wide variable
    4. global variable
12. When possible, you should avoid using \_\_\_\_\_\_\_\_\_\_ variables in a program.
    1. local
    2. global
    3. reference
    4. parameter
13. This is a prewritten function that is built into a programming language.
    1. standard function
    2. library function
    3. custom function
    4. cafeteria function
14. This standard library function returns a random integer within a specified range of values.
    1. random
    2. randint
    3. random\_integer
    4. uniform
15. This standard library function returns a random floating-point number in the range of 0.0 up to 1.0 (but not including 1.0).
    1. random
    2. randint
    3. random\_integer
    4. uniform
16. This standard library function returns a random floating-point number within a specified range of values.
    1. random
    2. randint
    3. random\_integer
    4. uniform
17. This statement causes a function to end and sends a value back to the part of the program that called the function.
    1. end
    2. send
    3. exit
    4. return
18. This is a design tool that describes the input, processing, and output of a function.
    1. hierarchy chart
    2. IPO chart
    3. datagram chart
    4. data processing chart
19. This type of function returns either True or False.
    1. Binary
    2. true\_false
    3. Boolean
    4. logical
20. This is a math module function.
    1. derivative
    2. factor
    3. sqrt
    4. differentiate
21. (True|False) The phrase “divide and conquer” means that all of the programmers on a team should be divided and work in isolation.
22. (True|False) Functions make it easier for programmers to work in teams.
23. (True|False) Function names should be as short as possible.
24. (True|False) Calling a function and defining a function mean the same thing.
25. (True|False) A flowchart shows the hierarchical relationships between functions in a program. – It’s called something else - hierarchy chart
26. (True|False) A hierarchy chart does not show the steps that are taken inside a function.
27. (True|False) A statement in one function can access a local variable in another function.
28. (True|False) In Python, you cannot write functions that accept multiple arguments.
29. (True|False) In Python, you can specify which parameter an argument should be passed into in a function call.
30. (True|False) You cannot have both keyword arguments and non-keyword arguments in a function call. – You can as long as non keyword arguments come first.
31. (True|False) Some library functions are built into the Python interpreter. – print, input, ect.
32. (True|False) You do not need to have an import statement in a program to use the functions in the random module. Import random
33. (True|False) Complex mathematical expressions can sometimes be simplified by breaking out part of the expression and putting it in a function.
34. (True|False) A function in Python can return more than one value. – You shouldn’t really though. It’s only really supposed to be doing one thing.
35. (True|False) IPO charts provide only brief descriptions of a function’s input, processing, and output, but do not show the specific steps taken in a function.
36. How do functions help facilitate teamwork? - allows you to divide and conquer the implementation of a program by assigning pieces of a program to various people.
37. Name and describe the two parts of a function definition. – header (def myfunction(0, 0): Body = rest of the owl
38. When a function is executing, what happens when the end of the function block is reached? Returns value if applicable, or just goes on to execute next function.
39. What is a local variable? What statements are able to access a local variable? – a variable scoped to a function. a value inside of structure.
40. What scope do parameter variables have? - Local scope to function
41. Why do global variables make a program difficult to debug? – because they can be changed anywhere
42. Suppose you want to select a random number from the following sequence: 0, 5, 10, 15, 20, 25, 30. What library function would you use? Randrange randrange(0, 31, 5)
43. What statement do you have to have in a value-returning function? – return
44. Draw an IPO chart that documents the input, processing, and output of the built-in input function. – no lol
45. What is a Boolean function? – a function that returns true or false often by using validation.
46. What are the advantages of breaking a large program into modules? – Easier to update, easier to break up the work and implement pieces.

Write a function named shout. The function should accept a string argument and display it in uppercase with an exclamation mark concatenated to the end. - AAAAAAAAAAAAA = input(str("SHOUT! :"))

print(AAAAAAAAAAAAA.upper(), "!", sep="")

1. The following statement calls a function named half, which returns a value that is half that of the argument. (Assume the number variable references a float value.) Write code for the function: result = half(number)