

COMPUTER SCIENCE

GE-3 (PYTHON PROGRAMMING)

FILL IN THE BLANKS

1. _____ is the first step in the software development process, where you identify and understand the problem to be solved. Answer: Problem definition
2. During _____, programmers create a plan or outline for how a program will solve a given problem. Answer: Program design
3. _____ is the process of identifying and removing errors or bugs in a computer program. Answer: Debugging
4. Syntax errors and logical errors are two common types of _____ in programming. Answer: Errors
5. Syntax errors occur when there are violations of the _____ rules of the programming language. Answer: Syntax
6. Logical errors, also known as _____ errors, occur when a program's logic is flawed, leading to incorrect results. Answer: Semantic
7. _____ is crucial in programming to make code understandable to others and your future self. Answer: Documentation
8. In documentation, comments are used to explain code _____ and provide context. Answer: Functionality
9. A well-documented program includes a clear _____ that describes its purpose, inputs, and outputs. Answer: Program header
10. _____ is the process of planning how a program will be structured and organized. Answer: Program design
11. The _____ is a critical part of program design, where you break down the problem into smaller, manageable tasks. Answer: Decomposition
12. Flowcharts and pseudocode are tools used in _____ to illustrate the program's logic. Answer: Program design
13. Debugging is often referred to as "_____", as it involves tracking down and fixing issues in the code. Answer: Bug hunting
14. A _____ is a specific test case that is designed to expose a particular error or problem in the program. Answer: Test case
15. In programming, _____ errors are easier to detect and fix than logical errors. Answer: Syntax
16. Logical errors can result from _____ in the program's algorithm or logic. Answer: Flaws
17. Good _____ helps future programmers understand the code and make necessary modifications. Answer: Documentation
18. Programmers use _____ to add explanations, notes, and comments within the code. Answer: Comments
19. The _____ of a program is a high-level description of what the program does and how it works. Answer: Program header
20. During _____, programmers plan the overall structure of the program and how various components will interact. Answer: Program design
21. _____ is the process of breaking down a complex problem into smaller, more manageable sub-problems. Answer: Decomposition
22. Flowcharts use _____ to represent program logic and flow. Answer: Symbols

23. Pseudocode is a _____ way of representing a program's logic using human-readable language. Answer: Simplified
24. Debugging involves identifying and correcting _____ in the code. Answer: Errors
25. Test cases are designed to _____ specific aspects of a program's functionality. Answer: Expose
26. Syntax errors result from violations of _____ rules. Answer: Language
27. Logical errors, also known as _____ errors, can lead to incorrect program behavior. Answer: Semantic
28. Proper _____ ensures that code is maintainable and can be easily updated or extended. Answer: Documentation
29. Comments within code provide _____ and explanations for the code's purpose and functionality. Answer: Context
30. The _____ of a program should include information about its purpose, inputs, expected outputs, and any relevant assumptions. Answer: Program header
31. _____ is a visual representation of a process that uses various symbols to illustrate the steps and decisions involved. Answer: Flowcharting
32. In flowcharts, a _____ is represented by an oval shape and indicates the start or end of a process. Answer: Terminal
33. A _____ is a graphical symbol used in flowcharts to represent a process or action. Answer: Rectangle
34. Decision points in a flowchart are represented by _____. Answer: Diamonds
35. A _____ is a systematic approach to solving a problem that specifies a sequence of steps to be followed. Answer: Algorithm
36. Algorithms are often written using a combination of _____ and natural language.
Answer: Pseudocode
37. _____ is a structured approach to software development that emphasizes the use of clear, modular, and organized code. Answer: Structured programming
38. In structured programming, _____ are used to break down a problem into smaller, manageable parts. Answer: Functions or procedures
39. The principle of _____ in structured programming states that a function should have a single, well-defined purpose. Answer: Cohesion
40. The principle of _____ in structured programming suggests that functions should communicate through well-defined interfaces. Answer: Coupling
41. In a _____ approach to problem solving, you start by breaking the problem into smaller sub-problems and solve them individually. Answer: Top-down
42. In _____ programming, you begin by solving the smallest sub-problems and then combine them to solve larger problems. Answer: Bottom-up
43. A decision table is a technique used to analyze complex decision-making scenarios and is represented as a _____. Answer: Table or matrix
44. Decision tables help in identifying _____ and their corresponding outcomes.
Answer: Conditions
45. In a decision table, rules are defined to specify the action or outcome based on combinations of _____. Answer: Conditions
46. A flowchart uses different shapes to represent various _____ in a process.
Answer: Elements or steps
47. The _____ shape in a flowchart represents a process or action to be performed.
Answer: Rectangle

48. A(n) _____ shape in a flowchart represents a decision point with two or more possible paths. Answer: Diamond
49. Structured programming encourages the use of _____ to avoid code duplication and improve code maintainability. Answer: Functions or procedures
50. In structured programming, _____ is the practice of dividing a complex problem into simpler, manageable modules. Answer: Modularization
51. In a top-down approach, you start by designing the _____ level of the program first. Answer: Highest or topmost
52. A _____ approach to problem solving involves solving individual sub-problems first and then combining them to solve the overall problem. Answer: Bottom-up
53. Decision tables are used to represent complex _____ scenarios in a tabular format. Answer: Decision-making
54. Flowcharts use various symbols and shapes to represent different _____ in a process. Answer: Steps or elements
55. In structured programming, high _____ between modules is discouraged. Answer: Coupling
56. A structured program is typically divided into _____ that perform specific tasks. Answer: Functions or procedures
57. Structured programming promotes code _____ and readability. Answer: Modularity
58. The top-down approach starts with designing the _____ of the program. Answer: Main structure or overview
59. In a bottom-up approach, you start by solving _____ sub-problems and build up to the complete solution. Answer: Individual or smaller
60. Decision tables help in organizing and simplifying complex _____ scenarios. Answer: Decision-making or logic
61. Python is an _____ programming language known for its simplicity and readability. Answer: Interpreted
62. In Python, a program is a sequence of _____ that the computer can execute. Answer: Instructions
63. Python uses _____ to define the structure of a program and indicate the beginning and end of blocks of code. Answer: Indentation
64. The Python interpreter is a program that reads Python _____ and executes them. Answer: Statements
65. You can use Python as a calculator by simply entering mathematical _____ in the Python shell. Answer: Expressions
66. The Python shell is an interactive environment where you can enter Python code and see the _____ immediately. Answer: Results
67. In Python, an _____ is a basic unit of data, such as numbers, text, or boolean values. Answer: Atom
68. An _____ is a name given to a variable, function, class, or other objects in Python. Answer: Identifier
69. Python has a set of reserved words, called _____, that cannot be used as identifiers. Answer: Keywords
70. Numeric literals in Python can be integers or _____ numbers. Answer: Floating-point
71. A _____ is a sequence of characters enclosed in single or double quotes. Answer: String

72. Python provides various operators, including _____ operators, for performing arithmetic operations. Answer: Arithmetic
73. The _____ operator in Python is used for addition, subtraction, multiplication, and division. Answer: Arithmetic
74. Relational operators are used to compare two values and return a _____ value. Answer: Boolean
75. The _____ operator in Python checks if two values are equal. Answer: Equality
76. Logical operators, also known as _____ operators, are used to combine boolean values. Answer: Boolean
77. The _____ operator in Python returns True if at least one of its operands is True. Answer: OR
78. The assignment operator (=) is used to assign a value to a _____. Answer: Variable
79. The ternary operator in Python is a shorthand way of writing an _____ statement. Answer: If-else
80. Bitwise operators are used for _____ manipulation of integers at the bit level. Answer: Binary
81. The _____ operator in Python is used to increment a variable's value by 1. Answer: Increment
82. The _____ operator in Python is used to decrement a variable's value by 1. Answer: Decrement
83. Python code is organized into _____, which are blocks of code that perform specific tasks. Answer: Functions
84. The _____ function is a built-in Python function used to display output to the console. Answer: print()
85. Python uses the _____ symbol (#) to indicate a comment in the code, which is not executed. Answer: Hash or Pound
86. The _____ statement is used for conditional execution of code based on a condition. Answer: if
87. In Python, the _____ statement is used for repetitive execution of code. Answer: for
88. A Python _____ is a collection of items that can be of different data types. Answer: List
89. Python provides a built-in _____ data type for storing key-value pairs. Answer: Dictionary
90. The _____ statement is used to define a function in Python. Answer: def
91. In Python, the _____ function is used to display text or values to the console. Answer: print()
92. The _____ function is used to take input from the user in Python. Answer: input()
93. Control statements in Python are used to manage the _____ of program execution. Answer: Flow
94. A _____ statement is used to make decisions in Python code based on a condition. Answer: Conditional
95. The _____ statement in Python is used to execute a block of code repeatedly as long as a condition is true. Answer: While
96. The _____ statement is used to iterate over a sequence, such as a list, tuple, or string. Answer: for
97. The _____ statement is used to exit a loop prematurely in Python. Answer: break

98. The _____ statement is used to skip the current iteration and continue to the next one in a loop. Answer: continue
99. The _____ statement is used as a placeholder for code that will be implemented later. Answer: pass
100. In Python, a function is defined using the _____ keyword. Answer: def
101. The values or variables that you pass to a function when calling it are called _____. Answer: Arguments
102. A function can have _____ parameters that are given default values. Answer: Default
103. Default arguments are used when a value is not provided for a _____ parameter in a function call. Answer: Default
104. In Python, you can return a value from a function using the _____ statement. Answer: return
105. The scope of a variable defined inside a function is _____ to that function. Answer: Local
106. A _____ function is a function that calls itself, often used for repetitive tasks. Answer: Recursive
107. The _____ function is used to exit the program and return a specified exit code. Answer: exit()
108. In a conditional statement, the code in the _____ block is executed when the condition is false. Answer: else
109. The _____ statement is used to check multiple conditions and execute different code blocks based on the conditions. Answer: elif
110. The _____ statement is used to check if a condition is true and execute a block of code accordingly. Answer: if
111. The _____ statement is used to check if a condition is true and execute a block of code if the condition is met. Answer: if
112. The _____ statement is used to repeat a block of code as long as a condition is true. Answer: while
113. The _____ statement is used to exit a loop or skip the current iteration. Answer: break
114. The _____ statement is used to continue to the next iteration of a loop. Answer: continue
115. The _____ statement is used as a placeholder for future code. Answer: pass
116. Default arguments are used to provide _____ values for function parameters. Answer: Default
117. You can override default arguments by providing _____ values when calling a function. Answer: Custom
118. The _____ statement is used to return a value from a function. Answer: return
119. A function can have both _____ and default parameters. Answer: Required
120. When using default arguments, parameters with defaults must appear _____ in the function's parameter list. Answer: Last

SHORT TYPE

Concept of Problem Solving:

1. What is problem-solving in programming?

- Problem-solving in programming is the process of designing algorithms or procedures to solve specific computational tasks or problems.

2. Why is problem definition important in programming?
- Problem definition is important because it helps in understanding the scope and requirements of a problem, which is essential for designing a solution.

Program Design:

3. What is program design?
- Program design is the process of planning the structure and organization of a computer program before writing the actual code.
4. Why is program design important?
- Program design helps in creating efficient and maintainable code by specifying the program's architecture and data flow.

Debugging:

5. What is debugging in programming?
- Debugging is the process of identifying and fixing errors or bugs in a computer program.
6. Name three common debugging techniques.
- Print statements, debugging tools, and code reviews.

Types of Errors in Programming:

7. What is a syntax error in programming?
- A syntax error occurs when the code violates the rules of the programming language.
8. What is a runtime error?
- A runtime error occurs while the program is executing and can lead to program crashes or unexpected behavior.
9. What is a logical error?
- A logical error is a type of error where the program runs without crashing but produces incorrect results due to flawed logic.

Documentation:

10. Why is documentation important in programming?
- Documentation helps other developers understand and use your code, making it easier to maintain and collaborate on projects.

Techniques of Problem Solving:

11. What is flowcharting?
- Flowcharting is a graphical representation of a process using various symbols to illustrate the sequence of steps.
12. What is a decision table?
- A decision table is a tabular method used to represent combinations of conditions and their corresponding actions.
13. What is an algorithm?
- An algorithm is a step-by-step procedure for solving a specific problem or task.

Structured Programming Concepts:

14. What is structured programming?
- Structured programming is a programming paradigm that emphasizes the use of structured control flow constructs like loops and conditionals.

Programming Methodologies:

15. What is top-down programming?
- Top-down programming is an approach where you start with a high-level overview of a program and then break it down into smaller sub-tasks.

16. What is bottom-up programming?

- Bottom-up programming is an approach where you build a program by first creating and testing smaller components or functions.

Overview of Programming:

17. What is the basic structure of a Python program?

- A Python program typically consists of import statements, variable declarations, function definitions, and the main program code.

18. What is the role of indentation in Python?

- Indentation is used in Python to define the structure and nesting of code blocks. It's crucial for readability and proper program execution.

Elements of Python:

19. What is a Python interpreter?

- A Python interpreter is a program that reads and executes Python code.

20. How can you use Python as a calculator?

- You can use Python's interactive shell to perform arithmetic operations like addition, subtraction, multiplication, and division.

21. What are literals in Python?

- Literals are constant values used in Python, such as numbers, strings, and Boolean values.

22. Give an example of a Python string literal.

- `"Hello, World!"`

23. What are Python keywords?

- Keywords are reserved words in Python that have special meanings and cannot be used as variable names.

24. Name a few Python keywords.

- `if`, `else`, `while`, `for`, `def`, `class`, `True`, `False`, `None`, etc.

Operators in Python:

25. What is an arithmetic operator?

- Arithmetic operators perform mathematical operations like addition, subtraction, multiplication, and division.

26. Give an example of a relational operator.

- `==` (equal to)

27. What do logical or Boolean operators do?

- Logical operators perform logical operations like AND, OR, and NOT on Boolean values.

28. What is the assignment operator in Python?

- `=` is used for assignment.

29. What is the ternary operator in Python?

- The ternary operator, `x if condition else y`, returns `x` if the condition is true, otherwise, it returns `y`.

30. Name a bitwise operator in Python.

- `&` (bitwise AND)

31. What do increment and decrement operators do?

- Python does not have explicit increment (`++`) and decrement (`--`) operators.

Creating Python Programs:

32. How can you take user input in Python?

- You can use the `input()` function to receive input from the user.

33. What are control statements in Python?
- Control statements are used to control the flow of a program, including branching and looping.
34. What is the purpose of a conditional statement?
- Conditional statements allow you to execute different blocks of code based on specified conditions.
35. What is the `exit()` function used for?
- The `exit()` function is used to exit the Python program with a specified exit code.
36. What is the difference between `break`, `continue`, and `pass` in Python?
- `break` is used to exit a loop prematurely, `continue` skips the rest of the current iteration and goes to the next, and `pass` is a placeholder statement with no effect.
37. What is a default argument in a function?
- A default argument is a parameter in a function that has a default value, and it's used when no value is provided by the caller.
38. How do you define a function in Python?
- You can define a function using the `def` keyword followed by the function name and parameters.

Input and Output Statements:

1. **Q:** How can you take user input in Python? **A:** You can use the `input()` function to receive input from the user.
2. **Q:** What function is used to display output in Python? **A:** The `print()` function is used to display output in Python.
3. **Q:** How do you print a variable's value along with a string? **A:** You can use f-strings or the `.format()` method to insert variable values into a string for printing.

Control Statements (Branching, Looping, Conditional Statements):

4. **Q:** What is branching in programming? **A:** Branching is the process of making decisions in code, often using `if` and `else` statements.
5. **Q:** What is a conditional statement? **A:** A conditional statement is used to execute code blocks based on a specified condition, such as `if` and `else`.
6. **Q:** What is the purpose of a `for` loop? **A:** A `for` loop is used to iterate over a sequence (e.g., a list) or execute a block of code a specific number of times.
7. **Q:** How is a `while` loop different from a `for` loop? **A:** A `for` loop is used for iterating over a sequence, while a `while` loop repeatedly executes a block of code as long as a specified condition is true.
8. **Q:** What does the `break` statement do in a loop? **A:** The `break` statement is used to exit a loop prematurely, stopping further iterations.
9. **Q:** How does the `continue` statement differ from `break`? **A:** The `continue` statement skips the rest of the current iteration and proceeds to the next one, whereas `break` exits the loop entirely.
10. **Q:** What is the purpose of the `pass` statement? **A:** The `pass` statement is a placeholder with no effect. It's often used when you need a code block with no action.
11. **Q:** How do you exit a Python program with a specific exit code? **A:** You can use the `exit()` function with the desired exit code as an argument, like `exit(0)`.

Defining Functions and Default Arguments:

12. **Q:** How do you define a function in Python? **A:** You can define a function using the `def` keyword, followed by the function name and parameters.

13. **Q:** What is a function's return statement used for? **A:** The `return` statement is used to specify the value that a function should return when called.
14. **Q:** What is a parameter in a function? **A:** A parameter is a variable used in a function's definition to receive input values when the function is called.
15. **Q:** What is an argument in a function call? **A:** An argument is the actual value or expression passed to a function when it's called, which corresponds to a function's parameter.
16. **Q:** What is a default argument in a function? **A:** A default argument is a parameter in a function that has a default value. If the caller doesn't provide a value for it, the default value is used.
17. **Q:** How do you specify default arguments in a Python function? **A:** Default arguments are specified in the function definition by assigning a default value to the parameter, like `def my_function(x=10):`.
18. **Q:** Can you override a default argument with a different value when calling a function? **A:** Yes, you can override a default argument by providing a different value when calling the function.
19. **Q:** What happens if you don't provide a value for a default argument when calling a function? **A:** If you don't provide a value for a default argument, the default value specified in the function definition is used.
20. **Q:** In Python, can a function have both parameters with default values and parameters without default values? **A:** Yes, a function can have a combination of parameters with default values and parameters without default values in its definition.

LONG TYPE

Problem Solving and Program Design

1. Explain the concept of problem-solving in programming.
2. What is problem definition, and why is it essential in programming?
3. Describe the stages involved in program design.
4. Discuss the significance of debugging in the software development process.
5. Enumerate and explain the various types of errors encountered in programming.
6. How does proper documentation impact the maintainability of a program?
7. Explain the techniques used in problem-solving in programming.
8. What is flowcharting, and how is it useful in program design?
9. Describe the purpose and construction of a decision table in programming.
10. Define algorithms and their role in solving computational problems.
11. Explain the key concepts of structured programming.
12. Compare and contrast top-down and bottom-up programming methodologies.
13. Provide an overview of the programming process, from problem identification to program execution.

Python Programming Concepts

14. What is the structure of a Python program, and why is it important?
15. Explain the fundamental elements of Python programming.
16. How does the Python interpreter work, and what is its role in program execution?
17. Discuss how Python can be used as a calculator with examples.
18. Describe the Python shell and its significance in interactive programming.
19. What is meant by indentation in Python, and why is it crucial?
20. Differentiate between atoms, identifiers, and keywords in Python.
21. Enumerate and explain various types of literals in Python.
22. Discuss different types of strings and their use in Python.

23. Provide an overview of Python operators and categorize them.
24. Explain the arithmetic operators in Python with examples.
25. Discuss relational operators in Python and their usage.
26. Describe the logical (Boolean) operators in Python and their applications.
27. How do assignment operators work in Python, and why are they important?
28. Explain the ternary operator in Python and its syntax.
29. Discuss bit-wise operators in Python and their use cases.
30. What is the significance of increment and decrement operators in Python?

Creating Python Programs

31. Describe the steps involved in creating Python programs.
32. How can you take user input in Python, and what are its applications?
33. Explain various methods for displaying output in Python programs.
34. Discuss branching control statements in Python, including if, elif, and else.
35. What is looping control, and how is it implemented in Python?
36. Differentiate between while and for loops in Python.
37. Explain conditional statements and their role in decision-making in Python.
38. How does the exit() function work in Python, and when is it used?
39. Compare and contrast the usage of break, continue, and pass statements in Python.
40. Define functions in Python and explain their advantages.

Default Arguments in Functions

41. What are default arguments in Python functions, and how are they defined?
42. Explain the role of default arguments in function parameters.
43. How do you override default arguments when calling a function?
44. Discuss scenarios where default arguments are beneficial in function design.
45. Provide examples of Python functions with default arguments.
46. Enumerate potential pitfalls when using default arguments in functions.