

QUESTION BANK

CLASS-9

2026-27

Chapter 1

Section A: Multiple Choice Questions

1. Computer instructions coded directly in terms of binary digits (0s and 1s) are known as:

- A) High-Level Language Code
- B) Assembly Language Code
- C) Machine Language Code
- D) Object-Oriented Code

2. The concept of mapping a single process or function name to handle multiple distinct sets of parameters is known as:

- A) Data Hiding
- B) Modular inheritance
- C) Method or Function Overloading
- D) Top-down flow sequencing

3. Java converts source program code into a highly secure, machine-independent intermediate binary code called:

- A) Mnemonics
- B) Op-codes
- C) Byte Code
- D) System script

4. Which standard Java package is imported automatically into every single Java application program by default?

- A) java.io.*
- B) java.util.*
- C) java.net.*
- D) java.lang.*

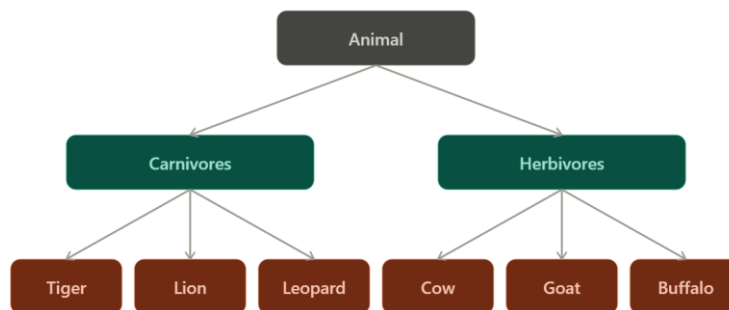
5. The non executable statement in java

- A) stand alone application
- B) print statement
- C) comment
- D) input statement

Section B: Picture-Based Reference Questions

Analyze the visual descriptions mapping directly to the chapters diagram pages and answer the evaluation boxes below.

1. Refer to the standard tree structure of classification mapping down from an "Animal" master box.



Question: Identify which blocks represent the Superclass/Base class and which represent the Subclass/Derived class. Define the principal concept that enables code reusability across these blocks.

Section C: Assertion and reason based

1. Assertion (A): A standalone Java application starts execution from the main() method, whereas a Java Applet relies on life-cycle methods such as init() and start() provided by the browser.

Reason (R): Standalone applications are executed by the JVM using the main method as the entry point, while applets are embedded in HTML and managed by the Java Plug-in.

- A. Both assertion and reason are true and reason is the correct explanation of assertion
- B. Both assertion and reason are true but reason is not the correct explanation of assertion
- C. Assertion is true but reason is false
- D. Assertion is false but reason is true

COMPREHENSIVE MARKING & ANSWER KEY

For Evaluators & Teachers Reference Only

Section A Answers (MCQs)

- 1. C
- 2. C
- 3. C

4. D

5. C

Section B Answers (Picture Evaluations)

• 1: 'Animal' represents the Super/Base class, while 'Carnivores' and 'Herbivores' act as Sub/Derived classes. Code reusability is driven by Inheritance.

Chapter 2: Elementary Concept of Objects and Classes

Section 1: Multiple Choice Questions (MCQs)

Q1. Which of the following is considered the absolute fundamental element and basic entity of Object-Oriented Programming?

- A. Class
- B. Object
- C. Method
- D. Data type

Q2. The characteristics of a real-world object correspond directly to which of the following in a software object?

- A. Member functions
- B. Data members
- C. Message passing
- D. Class factory

Q3. Which operator/keyword is utilized in Java to allocate space in dynamic memory for object storage?

- A. class
- B. public
- C. new
- D. static

Q4. A class is categorized as which type of data type in Java programming?

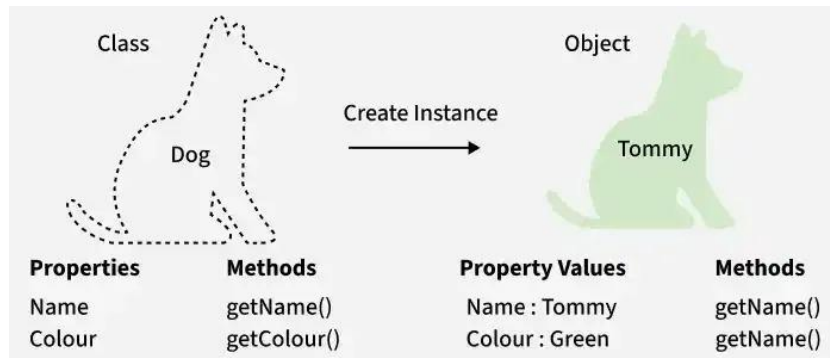
- A. Primitive Data Type
- B. Built-in Data Type
- C. User-Defined / Composite Data Type
- D. System Data Type

Q5. When objects communicate, exchange data, or invoke functions among each other, the process is explicitly known as:

- A. Data Abstraction
- B. Instantiation
- C. Message Passing
- D. Encapsulation

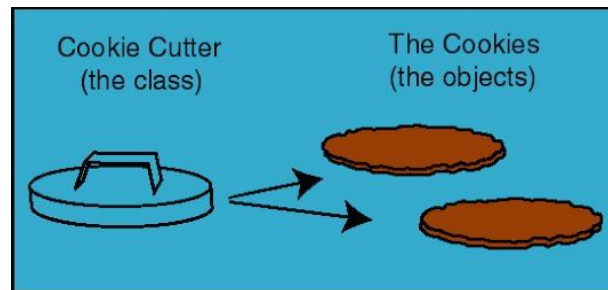
Section 2: Diagram & Picture-Based Analysis Questions

a)



- 1) Define a Class and an Object using the example given in the picture.
- 2) What is the process of creating an object from a class called?
- 3) Identify the two Properties (Attributes) defined in the Dog class.
- 4) Look closely at the Methods listed under the Object (Tommy). There is a mistake in the second method listed. Identify the mistake and write what it should be.

b)



Observe the given illustration and answer the following questions:

- 1) In Object-Oriented Programming (OOP), why is a Class often compared to a "Cookie Cutter"?
- 2) What do "The Cookies" represent in this analogy, and how do they relate to the cookie cutter?
- 3) Can you create multiple distinct cookies from a single cookie cutter? Relate this to a core property of

Section 3 Application – based question

A student is trying to write a Java program using a class named Student_Info to display a student's basic details. However, they left a few blanks in the code. Help them complete the program by answering the questions below.

```
java
```

```
class Student Info
```

```
{
```

```
void showName()
```

```
System.out.println("Ra
```

```
}
```

```
void showGrade()
{
System.out.println("C]
}
public static void main(St
Student Info student1
.showName(
student1.
}
```

Use code with caution.

Questions:

- (a) Name the keyword or access specifier that is most commonly used to make a method accessible from outside the class, to be filled in Line 1.
- (b) What element should be filled in the statement marked with Line 2 to complete the object creation?
- (c) What object reference name should be filled in the statement marked with Line 3 to invoke the showName() method?
- (d) What function/method name should be filled in the statement marked with Line 4 to print the student's grade details?

Section - 4 Assertion – reasoning

For each question, choose the correct option from the following uniform choices:

- (a) Both Assertion (A) and Reason (R) are true, and Reason (R) is the correct explanation of Assertion (A).
- (b) Both Assertion (A) and Reason (R) are true, but Reason (R) is not the correct explanation of Assertion (A).
- (c) Assertion (A) is true, but Reason (R) is false.
- (d) Assertion (A) is false, but Reason (R) is true

Assertion (A): Simply declaring a class variable like Car myCar; does not allocate memory for the object's instance variables.

Reason (R): Dynamic memory allocation for a Java object happens at runtime only when the new keyword is executed

Comprehensive Answer Key & Solution Explanations

Section:1 MCQ

- 1-b
- 2-b
- 3-c
- 4-c
- 5-c

Section 2: Diagram & Picture-Based Analysis Answers

Question 1 Solution:

- 1) **Class:** A blueprint or template that defines properties and behaviors (e.g., the `Dog` blueprint). **Object:** An identifiable entity or instance created from a class that has specific characteristics and behavior (e.g., `Tommy`, who is Green and has the name Tommy).
- 2) Instantiation (or "Creating an Instance").
- 3) Name and Colour.
- 4) The mistake is that `getName()` is repeated twice under the Object's methods. The second method should be `getColour()`.

Question 2 Solution:

Answer Key

1. A **Class** is called a "cookie cutter" because it acts as a blueprint, template, or factory design. It does not occupy memory space itself, but it determines the shape, structure, and features of the objects that will be created from it.
2. "The Cookies" represent the **Objects** (or instances). They are the actual physical entities created using the blueprint (class) that occupy memory space and possess real values.
3. **Yes.** A single class can be used to create any number of individual objects. Each object will share the same structure defined by the class, but can exist independently with its own distinct properties (e.g., different toppings or flavors for each cookie).

Section 3: application- based solution

(a) public: This access specifier ensures that the method can be accessed from any other class or the main method.

(b) Student_Info: The new keyword must be followed by the class constructor name (Student_Info()) to allocate memory for the new object.

(c) student1: To call an instance method inside main, you must use the identifier of the created object.

(d) showGrade: This invokes the second defined method in the class to output the grade information.

Section 4: assertion- reasoning

Correct Option: (a)Explanation: Declaration only creates a reference variable; the new keyword is required at runtime to allocate memory for the object.

Chapter-3

QUESTION 1: MULTIPLE CHOICE QUESTIONS.

1. What specific feature differentiates a 'String' data type from a primitive data type such as 'char' or 'boolean'?
 - (a) Strings store raw values directly
 - (b) Strings are mutable objects
 - (c) Strings store references / memory addresses
 - (d) Strings have no bit-width limit
2. What happens if an active variable is left uninitialised within a method block and subsequently evaluated inside an expression?
 - (a) It loads a default fallback value automatically
 - (b) It loads a random garbage value
 - (c) The compiler flags a compile-time mistake
 - (d) The platform breaks down at runtime
3. Which of the following choices specifies a valid 'null' literal initialization assignment pattern?
 - (a) String s = "null";
 - (b) String s = null;
 - (c) char c = null;
 - (d) int x = null;
4. In an expression involving 'char c', 'int i', and 'double d', what is the final resolution type reached via implicit type conversion logic?
 - (a) char
 - (b) int
 - (c) float
 - (d) double
5. What mathematical value is assigned to an evaluation of 'char c = 50;' when outputted directly as a character symbol?
 - (a) '0'
 - (b) '2'
 - (c) 'A'
 - (d) A blank empty block

QUESTION 2: CODE OUTPUT EVALUATION QUESTIONS

1.

```
char ch1 = '\u0041';
System.out.println("Character value = " + ch1);
```
2.

```
double result = 1 / 2 * 10.0;
System.out.println("Evaluated double matrix = " + result);
```
3.

```
double dValue = 45.99;
int iValue = (int) dValue;
System.out.println("Truncated target element = " + iValue);
```
4.

```
char marker = 'A';
System.out.println("Shift conversion index = " + (marker + 1));
```

5.
System.out.println("Character literal test check: " + (char)67);

QUESTION 3: ASSERTION AND REASON-BASED ANALYSIS.

Directions: Read the statements carefully. Choose option (a), (b), (c), or (d) according to the rule checklist:

- (a) Both Assertion and Reason are true, and Reason is the correct explanation for Assertion.
- (b) Both Assertion and Reason are true, but Reason is NOT the correct explanation for Assertion.
- (c) Assertion is true, but Reason is false.
- (d) Assertion is false, but Reason is true.

- 1) **Assertion (A):** Explicit type conversion (typecasting) can result in data loss.
Reason (R): It often involves converting a larger data type into a smaller data type.
- 2) **Assertion (A):** The character combination \n is considered a single character when evaluated inside a string literal.
Reason (R): Escape sequences always begin with a backslash (\) followed by a specific formatting character, and the compiler interprets the combined pair as a single control character.

QUESTION 4: APPLICATION-BASED ENGINEERING SCENARIO

- 1) In Java, every small individual unit in a program is known as a token. The compiler recognizes these tokens to parse and understand the code. Among these tokens, names given to different program elements like variables, methods, and classes are called identifiers. Java enforces strict rules for naming these identifiers; for instance, they cannot begin with a digit, and they are case-sensitive. Additionally, Java reserves certain words for its own internal syntax, which have predefined meanings and cannot be used as identifiers. To store actual, fixed data values directly in a program, programmers use literals, which can represent integers, floating-point numbers, characters, or boolean states.

Based on the above discussion, answer the following questions:

(a) What is the term used for the smallest individual units in a Java program?

- 1) Literals
- 2) Tokens
- 3) Identifiers
- 4) Variables

(b) Reserved words with predefined meanings in Java that cannot be used as variable names are known as:

- (i) Keywords
- (ii) Constants
- (iii) System names
- (iv) Identifiers

(c) Which of the following is a valid identifier name according to Java rules?

- (i) 2ndPlace (*Invalid: starts with a digit*)
- (ii) total_score (*Valid: uses alphanumeric characters and underscore*)
- (iii) class (*Invalid: it is a reserved keyword*)
- (iv) grand-total (*Invalid: contains a hyphen*)

(d) What term is used to describe fixed, constant values directly assigned to a variable in the source code?

- (i) Operators
- (ii) Separators

- (iii) Literals
- (iv) Identifiers

QUESTION 5: TYPE CONVERSION & CASTING MATRIX .

Directions: Determine the implicit target variable data types or execute explicit adjustments for the following execution contexts.

1. Rewrite the expression using explicit type casting to force the calculation to evaluate as an int value type:
float f; char c; short s;
Base Formula: $f + c / s$;
2. Rewrite the target statement using manual explicit type casting to force the entire evaluated outcome to convert into a float configuration:
int a; float f; double d; char c;
Base Formula: $(a / f) - d * c$;

ANSWER KEY

QUESTION 1: MULTIPLE CHOICE ANSWERS

- 1- (c) Strings store references / memory addresses
- 2- (c) The compiler flags a compile-time mistake
- 3- (b) String s = null;
- 4- (d) double
- 5- (b) '2'

QUESTION 2: EXPECTED OUTPUT KEY CODES

- 1- Character value = A
- 2- Evaluated double matrix = 0.0
- 3- Truncated target element = 45
- 4- Shift conversion index = 66
- 5- Character literal test check: C

QUESTION 3: ASSERTION REASON RESOLUTION

- 1) **Correct Answer: (a) > Explanation:** Because a constant cannot be reassigned a value later in the program, it must be given its initial value right at the moment it is declared. Therefore, both are true, and R correctly explains A.
- 2) **Correct Answer: (a) Explanation:** Because non-primitive types can dynamically grow and hold complex multi-value structures, the runtime allocation manages them on the heap and handles them via fixed-size memory pointers (references) on the stack.

QUESTION 4: APPLICATION ENGINEERING CODE CORRECTION

- 1) (a) - (ii) Tokens (b) - (i) Keywords (c) - (ii) total_score (d) - (iii) Literals

QUESTION 5: TYPE CONVERSION ANSWERS

1. (int) $(f + c / s)$
2. (float) $((a / f) - d * c)$

Chapter -4

SECTION 1: MULTIPLE CHOICE QUESTIONS.

1. Which type of operator evaluates precisely three distinct operand expressions to achieve conditional structural assignments?

- (a) Unary Operator
- (b) Binary Operator
- (c) Ternary Operator
- (d) Shorthand Assignment Operator

2. If an execution expression contains all three primary logical operators (!, &&, ||), what is their legal sequence of precedence evaluation from left to right?

- (a) && followed by || followed by !
- (b) ! followed by && followed by ||
- (c) || followed by && followed by !
- (d) Evaluated purely left-to-right regardless of structural symbols

3. What criteria differentiates a 'Counter' numeric variable from an 'Accumulator' numeric variable within iterative execution blocks?

- (a) Counters decrement while accumulators can only increment
- (b) A counter variable tracks process iterations by updating via a constant scale (typically 1), while an accumulator stores running totals updated by dynamic values
- (c) Counters store textual data patterns, whereas accumulators handle primitive binary logic states
- (d) Counters exist entirely within class templates; accumulators exist only in method parameter streams

4. According to Java's formal hierarchy matrix of operators, which pair of basic operational symbols shares the exact same precedence tier?

- (a) +, *
- (b) *, %
- (c) &&, ||
- (d) =, ==

5. What specific initialization value must be applied to an accumulator variable configuration to safely guard against compiling garbage elements?

- (a) It must be initialized to 1
- (b) It must be left completely uninitialized to optimize dynamic scope step metrics
- (c) It should be initialized to a controlled base value (such as 0 for summations) to anchor operational processing steps

(d) It must be bound with the new storage modifier keyword string

6. What syntax token functions explicitly as the structural 'connector' or 'concatenation' symbol to join variables and strings inside print parameters?

- (a) The dot operator symbol (.)
- (b) The plus symbol (+)
- (c) The comma separator tag (,)
- (d) The colon logical token symbol (:)

SECTION 2: CODE OUTPUT EVALUATION

Directions: Carefully analyze the runtime operation loops, execution hierarchies, and formatting characteristics of each unique isolated segment below. Write down the precise console string sequence generated.

1.

```
int a = 48;
a = a++ + ++a;
System.out.println("a = " + a);
```

2.

```
int c = 2;
int d = ++c + c++ + 4;
System.out.println("d = " + d);
System.out.println("c = " + c);
```

3.

```
int a = 8;
a -= ++a + a++ + 4;
System.out.println("a = " + a);
```

4.

```
int a = 12, b = 8;
a *= ++a / 6 + b++ % 3;
System.out.println("a = " + a);
```

5.

```
int a = 22, b = 5;
System.out.println("Quotient=" + (a / b));
System.out.println("Remainder=" + (a % b));
```

6.

```
System.out.println("India won ICC Cricket World Cup");
System.out.print("against ");
System.out.println("West Indies");
System.out.print("in the year 1983");
```

7.

```
boolean flag = (5 > 3) && (3 < 0) || !(4 == 2);
System.out.println("Resulting Flag Matrix = " + flag);
```

SECTION 3: ASSERTION & REASON ANALYTICAL MATRIX

Directions: Read the targeted statement premises below attentively. Select the appropriate classification choice according to the checklist logic definitions:

- (a) Both Assertion and Reason match structural truth values, and Reason is the clear systemic explanation for the Assertion line.
- (b) Both Assertion and Reason match factual values, but Reason is NOT the target mechanism explanation path.
- (c) Assertion is correct, but the stated Reason line is factually flawed.
- (d) Assertion is false, but the stated Reason line is correct.

Assertion (A): The + operator can be used to join two or more string variables together in Java. Reason (R): The + operator changes its behavior based on the data types of its operands, allowing it to perform string concatenation in addition to arithmetic addition.

SECTION 4: APPLICATION-BASED DEVELOPMENT SCENARIO

Java language utilizes distinct syntax rules, specialized keywords, and structural operators to control object initialization, manage variable scopes, and establish conditions. Evaluation of these components relies heavily on strict operator precedence rules (like the Java operator hierarchy) rather than basic mathematics. Base on the above discussion, answer the following questions:

- (a) Which keyword is used to initialize an instance variable when it has the same name as a local variable within a constructor or method?
 - (i) new
 - (ii) this
 - (iii) super
 - (iv) static
- (b) Which of the following operator groups have the same precedence level in Java?
 - (i) + and ++
 - (ii) * and %
 - (iii) && and ||
 - (iv) == and =
- (c) _____ operator is used to check whether an object belongs to a specific class or interface type.
 - (i) instanceof
 - (ii) extends
 - (iii) implements
 - (iv) . (dot)
- (d) What will be the correct precedence order of relational and equality operators (from highest to lowest)?
 - (i) ==, !=, >, <
 - (ii) >, <, ==, !=
 - (iii) !=, ==, <, >
 - (iv) All have the exact same precedence

SECTION 5: TYPE CONVERSION & AUTOMATIC PROMOTION MATRIX

1. Context Scenario:

```
int a; float f; char c; double d;
Expression outcome type target query: a + f / c * d;
```

2. Context Scenario:

```
short s; byte b; int i;  
Expression outcome type target query: s * b - i;
```

3. Context Scenario:

```
char ch = 'A'; // ASCII 65  
int value = ch + 5;  
What is the value and explicit storage type profile of 'value'?
```

4. Context Scenario:

```
double dValue = 145.82;  
Code Statement: int truncatedInt = (int) dValue;  
Determine the explicit result stored inside truncatedInt.
```

5. Context Scenario:

```
int p = 25;  
Code Statement: float resultFloat = (float) p / 2;  
Determine the dynamic numeric data evaluated within resultFloat.
```

SECTION 6: CONVERTING MATHEMATICAL EXPRESSIONS TO JAVA SYNTAX

- 1. Mathematical Expression Format:** abc (Product sequence layout mapping)
Convert this expression syntax into a clean, compilation-ready Java expression statement.
- 2. Mathematical Expression Format:** $ab - bc + ca$ (Compound multi-variable group)
Convert this expression syntax into a clean, compilation-ready Java expression statement.
- 3. Mathematical Expression Format:** $a^2 + b^2 - c^2$ (Exponential geometric calculation configuration)
Convert this expression syntax into a clean, compilation-ready Java expression statement.
- 4. Mathematical Expression Format:** $2(l + b)$ (Standard perimeter multiplication framework)
Convert this expression syntax into a clean, compilation-ready Java expression statement.
- 5. Mathematical Expression Format:** $(P * r * t) / 100$ (Standard financial yield ratio tracing configuration)
Convert this expression syntax into a clean, compilation-ready Java expression statement.

SECTION 7: CONVERTING JAVA SYNTAX EXPRESSIONS TO STANDARD MATHEMATICS

- $1.0 / 3.0 * a * b + 1.0 / 2.0 * c * d$
- $b * b - 4 * a * c$
- $a * a * a - 3 * \text{Math.pow}(a,20) / \text{Math.sqrt}(b) - c * c$
- $\text{Math.sqrt}(2) * s$
- $cp * 20.0 / 100.0$

COMPREHENSIVE SYSTEMS ANSWER KEY & STRUCTURAL RESOLUTIONS

Provides verification indexes, explicit processing steps, console character trace outputs, and derivation details for the examination materials.

=== SECTION 1: MULTIPLE CHOICE CORRECT CHOICES ===

- 1- (c) Ternary Operator
- 2- (b) ! followed by && followed by ||
- 3- (b) A counter variable tracks process iterations by updating via a constant scale (typically 1), while an accumulator stores running totals updated by dynamic values
- 4- (b) *, %
- 5- (c) It should be initialized to a controlled base value (such as 0 for summations) to anchor operational processing steps
- 6- (b) The plus symbol (+)

=== SECTION 2: EXPECTED CONSOLE OUTPUT TRACE LIST ===

- 1) a = 98
- 2) d = 11
c = 4
- 3) a = -13
- 4) a = 36
- 5) Quotient=4
Remainder=2
- 6) India won ICC Cricket World Cup against West Indies in the year 1983
- 7) Resulting Flag Matrix = true

=== SECTION 3: ASSERTION & REASON ANALYTICAL RESOLUTION ===

Correct Option Selection: (a)

=== SECTION 4: APPLICATION DEVELOPMENT SOLUTION ===

Application based questions

- (a) (ii) new
- (b) (iii) * and /
- (c) (i) . (dot)
- (d) (iii) !, && and ||

=== SECTION 6: CONVERTED MATHEMATICAL JAVA EXPRESSIONS ===

Java Expression Translation 1: $a * b * c$

Java Expression Translation 2: $a * b - b * c + c * a$

Java Expression Translation 3: $a * a + b * b - c * c$

Java Expression Translation 4: $2 * (1 + b)$

Java Expression Translation 5: $(P * r * t) / 100$

=== SECTION 7: CONVERTED JAVA TO MATHEMATICAL EXPRESSIONS ===

Algebraic Formula Translation 1: $(1/3)ab + (1/2)cd$

Algebraic Formula Translation 2: $b^2 - 4ac$

Algebraic Formula Translation 3: $a^3 - 3a^2b$

Algebraic Formula Translation 4: $\sqrt{2} \times s$ (Diagonal tracking scaling)

Algebraic Formula Translation 5: 20% of cp (Discount value mapping formula)