

Variables, Expressions

Multiple choice questions

- Where do we declare local variables ?
 - It declares inside the class and outside of method.
 - It declares inside the class with static prefix and outside of method.
 - It declares inside the method.
 - none of these
- Where do we declare instance variables ?
 - It declares inside the class and outside of method.
 - It declares inside the class with static prefix and outside of method.
 - It declares inside the method.
 - none of these
- Where do we declare class variables ?
 - It declares inside the class and outside of method.
 - It declares inside the class with static prefix and outside of method.
 - It declares inside the method.
 - none of these
- Which element can not be used as variable name in Java programming language ?
 - literal
 - Keywords
 - Identifier
 - Identifier and Keywords
- Name of variable is called:
 - Data Type
 - Constant
 - Identifier
 - None of the above
- Which variable name is invalid variable name ?
 - tenthTeam
 - TenthTeam
 - 10thTeam
 - None of the Above
- A variable name should never begin with.
 - A number
 - Special characters but underscore and dollar sign
 - All of the above
- Choose a valid variable name.
 - 5thTeam
 - &5thteam
 - _5thTeam
 - None of the Above
- Which variable names are the invalid variable names ?
 - 1stLevel
 - first Level
 - *firstLevel
 - All of the Above
- Which variable names are the valid variable names ?
 - \$1stLevel
 - _1stLevel
 - £1stLevel
 - All of the Above
- The two possible Logical Operator types are:
 - Bitwise Logical
 - Boolean Logical
 - Arithmetic Logical
 - (a) and (b)
- Which data type do we use with Boolean logical operators in Java ?
 - true/false boolean data
 - 1 and 0 of individual Bits
 - characters of a String
 - None of the above
- Which type of data can be used with Bitwise logical operators in Java?
 - true/false boolean data
 - 0 and 1 individual bits of data
 - Characters of a String
 - None of the above
- Bitwise logical operators are also called _____.
 - Logical operators
 - Bitwise operators
 - Binary operators
 - None of the above
- Input used for Logical Operators are:
 - 1 and 0
 - true / false
 - char / String
 - None of the above

16. Output given by any Logical operation in Java:
 - (a) 1 or 0
 - (b) true or false
 - (c) char or String
 - (d) None of the above
17. Which Logical operator works with a Single Operand?
 - (a) Logical AND
 - (b) Logical OR
 - (c) Logical Exclusive OR
 - (d) Logical NOT
18. Which is a Logical Unary NOT operator in Java?
 - (a) ~
 - (b) !
 - (c) #
 - (d) ^
19. What will be the output of a Logical OR (|) operation if one of the inputs/operands is false?
 - (a) false
 - (b) true
 - (c) true or false
 - (d) None of the above
20. What will be the output of Logical AND (&) operation if one of the inputs/operands is false?
 - (a) false
 - (b) true
 - (c) true or false
 - (d) None of the above
21. What will be the output for a Logical OR (|) operation when inputs/operands is true?
 - (a) false
 - (b) true
 - (c) true or false
 - (d) None of the above
22. What will be the output of a Logical AND (&) operation when inputs/operands is true?
 - (a) false
 - (b) true
 - (c) true or false
 - (d) None of the above
23. What will be the output of a Logical AND (&) operation if both inputs/operands are true?
 - (a) false
 - (b) true
 - (c) true or false
 - (d) None of the above
24. What will be the output of a Logical OR (|) operation if both the inputs/operands are true?
 - (a) true
 - (b) false
 - (c) true or false
 - (d) None of the above
25. Which operator is fast AND (&) and Short Circuit AND(&&) operators in Java?
 - (a) AND operator
 - (b) Short Circuit AND
 - (c) Both work at the same speed
 - (d) None of the above
26. Which operator is fast OR(|) and Short Circuit OR (||) operators in Java?
 - (a) OR Operator
 - (b) Short Circuit OR operator
 - (c) Both work at the same speed
 - (d) None of the above
27. Why Short Circuit AND (&&) and Short Circuit OR (||) operators are fast in Java?
 - (a) By skipping the second expression or operand if possible and save time.
 - (b) By using extra memory on the machine
 - (c) By using extra CPU processing power
 - (d) None of the above
28. Which operators are involved in Arithmetic expression in Java ?
 - (a) Addition (+), Subtraction (-)
 - (b) Multiplication (*), Division (/)
 - (c) Modulo Division (%), Increment/Decrement (++/--), Unary Minus (-), Unary Plus (+)
 - (d) All of the above
29. Which is the correct Compound Assignment Arithmetic Operators in Java .
 - (a) +=, -=
 - (b) *=, /=
 - (c) %=
 - (d) All of the above
30. Choose the correct output of Java code snippet?


```
int a = 2 - - 7;
System.out.println(a);
```

 - (a) -5
 - (b) 10
 - (c) 9
 - (d) Compiler Error
31. Choose the correct output of Java code snippet ?


```
short p = 1;
short k = p + 2;
System.out.println(k);
```

 - (a) 1
 - (b) 2
 - (c) 3
 - (d) Compiler error
32. The arithmetic operator in Java that gives the Remainder of Division is :
 - (a) /
 - (b) @
 - (c) %
 - (d) &
33. What type of associativity do Arithmetic operators +, -, /, * and % have?
 - (a) Right to Left
 - (b) Left to Right
 - (c) Right to Right
 - (d) Left to Left
34. Which of the following operators have more priority?
 - (a) Postfix operators have more priority than Prefix operators
 - (b) Prefix operators have more priority than Postfix operators
 - (c) Both Prefix and Postfix operators have equal priority
 - (d) None of the above

35. Which of the following operator has less priority?
 (a) Postfix Decrement has less priority than Prefix Increment
 (b) Prefix Increment has less priority than Postfix Decrement
 (c) Both operators have same priority
 (d) None of the above
36. How is the associativity used by Increment and Decrement arithmetic operators in Java ?
 (a) Left to Right (b) Right to Left
 (c) Left to Left (d) Right to Right
37. Which of the following is the correct statement about Java Operators +, -, *, / and %.
 (a) + and - have equal priority
 (b) * and / have equal priority
 (c) / and % have equal priority
 (d) All the above
38. Choose the group with higher priority in operator groups (++ , --) and (+, -, *, /, %) in Java.
 (a) (++ , --) group has higher priority than (+, -, *, /, %) group
 (b) (++ , --) group has lower priority than (+, -, *, /, %) group
 (c) (++ , --) group and (+, -, *, /, %) group have equal priority
 (d) None of the above
39. Operator with highest precedence:
 (a) () (b) ++
 (c) * (d) >>
40. _____ is the order of precedence (highest to lowest) of following operators?
 1. &
 2. ^
 3. ?:
 (a) 1 -> 2 -> 3 == (b) 2 -> 1 -> 3
 (c) 3 -> 2 -> 1 (d) 2 -> 3 -> 1
41. _____ statement is incorrect?
 (a) Equal to operator has least precedence
 (b) Brackets () have highest precedence
 (c) Division operator, /, has higher precedence than multiplication operator
 (d) Addition operator, +, and subtraction operator have equal precedence
42. _____ returned by greater than, <, and equal to, ==, operator?
 (a) Integers
 (b) Floating - point numbers
 (c) Boolean
 (d) None of the mentioned
43. _____ operators can operate on a boolean variable?
 1. &&
 2. ==
 3. ?:
 4. +=
 (a) 3 & 2 (b) 1 & 4
 (c) 1, 2 & 4 (d) 1, 2 & 3
44. _____ operators can skip evaluating right hand operand.
 (a) ! (b) |
 (c) & (d) &&
45. _____ operator is having highest precedence.
 (a) () (b) ++
 (c) * (d) >>
46. Expression1 should be _____ to evaluate using ternary operator?
 expression1 ? expression2 : expression3
 (a) Integer
 (b) Floating – point numbers
 (c) Boolean
 (d) None of the mentioned
47. _____ is the value stored in x in following lines of code?

```
int x, y, z;
x = 0;
y = 1;
x = y = z = 8;
```

 (a) 0 (b) 1
 (c) 9 (d) 8
48. _____ is the value stored in x in following lines of code?

Fill in the blanks

40. _____ should be expression1 evaluate to in using ternary operator in the following line?
 expression1 ? expression2 : expression3
 (a) Integer
 (b) Floating – point numbers
 (c) Boolean
 (d) None of the above
41. _____ is the value stored in x in the following lines of Java code?

```
int x, y, z;
x = 0;
y = 1;
x = y = z = 8;
```

 (a) 0 (b) 1
 (c) 9 (d) 8
42. _____ is the value stored in x in following lines of code?

```
int x, y, z;
x = 0;
y = 1;
x = y = z = 8;
```

- (a) 0 (b) 1
(c) 9 (d) 8

Match the following

50. Match the columns:

Group A	Group B
1. Operands of arithmetic operators	(i) Integers and floating - point numbers
2. Modulus operator, %,	(ii) Numeric & Characters
3. Operator, ^~, decreases the value	(iii) () , { }
4. Highest order precedence operator	(iv) by 1

- (a) 1-(ii), 2-(i), 3-(iv), 4-(iii)
(b) 1-(i), 2-(ii), 3-(iv), 4-(iii)
(c) 1-(ii), 2-(i), 3-(iii), 4-(iv)
(d) 1-(ii), 2-(iv), 3-(i), 4-(iii)

51. Match the columns:

Group A	Group B
1. && and operators	(i) Ternary Operator
2. Colon (?) operator	(ii) Combine two boolean values
3. Java Ternary operator	(iii) true or false
4. Condition of a Java Ternary operator	(iv) Conditional Operator

- (a) 1-(ii), 2-(i), 3-(iv), 4-(iii)
(b) 1-(i), 2-(ii), 3-(iv), 4-(iii)
(c) 1-(ii), 2-(i), 3-(iii), 4-(iv)
(d) 1-(ii), 2-(iv), 3-(i), 4-(iii)

52. Match the columns:

Group A	Group B
1. Assignment operator	(i) must return a value
2. True part of conditional operator	(ii) Assignment and Lambda operator

3. False Part of conditional operator	(iii) \ \
4. Character escape code not in java	(iv) Must return a value

- (a) 1-(ii), 2-(i), 3-(iv), 4-(iii)
(b) 1-(i), 2-(ii), 3-(iv), 4-(iii)
(c) 1-(ii), 2-(i), 3-(iii), 4-(iv)
(d) 1-(ii), 2-(iv), 3-(i), 4-(iii)

53. Match the columns:

Group A	Group B
1. Portability and security of Java	(i) Use of pointers
2. Not a Java features	(ii) Byte code is executed by JVM
3. \u0021 article	(iii) JDBC
4. Find and fix bugs	(iv) Unicode escape sequence

- (a) 1-(ii), 2-(i), 3-(iv), 4-(iii)
(b) 1-(i), 2-(ii), 3-(iv), 4-(iii)
(c) 1-(ii), 2-(i), 3-(iii), 4-(iv)
(d) 1-(ii), 2-(iv), 3-(i), 4-(iii)

54. Match the columns:

Group A	Group B
1. Return type of the hash Code() method	(i) 0xnf029L
2. Valid long literal	(ii) int
3. Float a = 35 / 0 return?	(iii) It has no class name
4. Anonymous inner class	(iv) Infinity

- (a) 1-(ii), 2-(i), 3-(iv), 4-(iii)
(b) 1-(i), 2-(ii), 3-(iv), 4-(iii)
(c) 1-(ii), 2-(i), 3-(iii), 4-(iv)
(d) 1-(ii), 2-(iv), 3-(i), 4-(iii)

Programming based questions

55. Choose correct output of Java code snippet?

```
short k=1;
```

```
k += 2;
```

```
System.out.println(k);
```

- (a) 1
(b) 2
(c) 3
(d) Compiler error about Type Casting

56. With reference to the program code given below, answer the questions that follow:

```
public class ReverseNumber
{
    public static void main(String[] args)
    {
        int number = 987654, reverse = 0;
        while(number != 0)
        {
            int remainder = number % 10;
            reverse = reverse * 10 + remainder;
            number = number/10;
        }
        System.out.println("The reverse of the
        given number is: " + reverse);
    }
}
```

- (a) What is the output of the following program?
- The reverse of the given number is: 456789
 - Compilation error
 - 15
 - Runtime error
- (b) What is the name of the class in the above program
- Constructor
 - ReverseNumber
 - String[] args
 - Name

57. Regarding the program code given below, answer the questions that follow:

```
public class SumOfNaturalNumber2
{
    public static void main(String[] args)
    {
        int num = 100, i = 1, sum = 0;
        //executes until the condition returns true
        while(i<= num)
        {
            //adding the value of i into sum variable
            sum = sum + i;
            //increments the value of i by 1
            i++;
        }
        //prints the sum
        System.out.println("Sum of First 100
```

```
Natural Numbers is = " + sum);
    }
}
```

- (a) What is the output of the following program?
- Sum of First 100 Natural Numbers is = 5050
 - Compilation error
 - 15
 - Runtime error
- (b) What is the name of the class in the above program ?
- Constructor
 - Sum of Natural Number 2
 - String[] args
 - Name

58. Regarding the program code given below, answer the questions that follow:

```
public class CheckPositiveOrNegative
{
    public static void main(String[] args)
    {
        //number to be check
        int num=912;
        //checks the number is greater than 0 or not
        if(num>0)
        {
            System.out.println("The    number    is
            positive.");
        }
        //checks the number is less than 0 or not
        else if(num<0)
        {
            System.out.println("The    number    is
            negative.");
        }
        //executes when the above two conditions
        return false
        else
        {
            System.out.println("The number is zero.");
        }
    }
}
```

(a) What is the output of the following program?

- (i) The number is positive.
- (ii) The number is Negative.
- (iii) 15
- (iv) Runtime error

(b) What does the above program do?

- (i) Initialize a number
- (ii) Check whether the number is positive or negative
- (iii) Give odd or even number
- (iv) None of the above

Answers

Multiple choice questions

1. (c) It declares inside the method.

Explanation: A variable declared inside the body of the method is called local variable. You can use this variable only within that method and the other methods in the class aren't even aware that the variable exists. A local variable cannot be defined with "static" keyword.

2. (a) It declares inside the class and outside of method.

Explanation: Instance variables are declared in a class, but outside a method, constructor or any block. When space is allocated for an object in the heap, a slot for each instance variable value is created.

3. (b) It declares inside the class with static prefix and outside of method.

Explanation: Class variables also known as static variables are declared with the static keyword in a class, but outside a method, constructor or a block. There would only be one copy of each class variable per class, regardless of how many objects are created from it.

4. (b) Keywords

Explanation: Java keywords are also known as reserved words. Keywords are particular words that act as a key to a code.

5. (c) Identifier

Explanation: Identifiers in Java are symbolic names used for identification. They can be a class name, variable name, method name, package name, constant name, and more.

6. (d) None of the Above

7. (c) All of the above

8. (c) _5thTeam

9. (d) All of the Above

10. (d) All of Above

11. (d) A and B

12. (a) true/false boolean data

13. (b) 0 and 1 individual bits of data

14. (b) Bitwise operators

Explanation: Java defines several bitwise operators, which can be applied to the integer types, long, int, short, char, and byte.

15. (b) true / false

16. (b) true or false

17. (d) Logical NOT

18. (b) !

19. (c) true or false

20. (a) false

21. (b) true

22. (c) true or false

23. (b) true

24. (a) true

25. (b) Short Circuit AND

Explanation: In Java logical operators, if the evaluation of a logical expression exits in between before complete evaluation, then it is known as Short-circuit. ... If there is an expression with &&(logical AND), and first operand itself is false, then short circuit occurs, the further expression is not evaluated and false is returned.

26. (b) Short Circuit OR operator

27. (a) By skipping the second expression or operand if possible and save time.

28. (d) All of the above

29. (d) All of the above

30. (c) 9

31. (d) Compiler error

32. (c) %

33. (b) Left to Right

34. (a) Postfix operators have more priority than Prefix operators

35. (b) Prefix Increment has less priority than Postfix Decrement

36. (b) Right to Left
37. (d) All of the above
38. (a) (++ , --) group has higher priority than (+, -, *, /, %) group
39. (a) ()

Fill in the blanks

40. (c) Boolean
41. (d) 8
42. (a) 1 -> 2 -> 3 ==
43. (c) Division operator, /, has higher precedence than multiplication operator
44. (c) Boolean
45. (d) 1, 2 & 3
46. (d) &&
47. (a) ()
48. (c) Boolean
49. (d) 8

Match the following

50. (a) 1-(ii), 2-(i), 3-(iv), 4-(iii)
51. (a) 1-(ii), 2-(i), 3-(iv), 4-(iii)

52. (a) 1-(ii), 2-(i), 3-(iv), 4-(iii)
53. (a) 1-(ii), 2-(i), 3-(iv), 4-(iii)
54. (a) 1-(ii), 2-(i), 3-(iv), 4-(iii)

Programming based questions

55. (c) 3
56. (a) (i) The reverse of the given number is: 456789
(b) (ii) ReverseNumber
57. (a) (ii) Compilation error

Explanation: Constructors can be chained and overloaded. When Test() is called, it creates another Test object calling the constructor Test(int temp).

- (b) (ii) Sum of natural Number 2
58. (a) (i) The number is positive.
(b) (ii) Check whether the number is positive or negative