OOP Questions in Java (Solutions)

Collected by:
Omar AbdulRahman

1. Output: Compiler Error
   Reason: There is an error in condition check expression of for loop. Java differs from C++(or C) here. C++ considers all non-zero values as true and 0 as false. Unlike C++, an integer value expression cannot be placed where a boolean is expected in Java.

2. Output: Compiler Error
   Reason: Unlike C++, static local variables are not allowed in Java. The correct way is to declare it in the beginning outside any methods.

3. Answer: b) Error
   Reason: Break statement can only be used with loop or switch. So, using break with if statement causes “break outside switch or loop” error.

4. Answer: b) Something else (Other than simple concatenation)
   Reason: “java” would be printed if String literals (in double quotes) are used, but in the question since character literals has been used, these won’t be concatenated. Therefore, After execution of the program, an addition of each equivalent ASCII(Unicode) value of the character will be obtained.
   Hence the output is 106 + 97 + 118 + 97 = 418

5. Answer: a) Nothing
   Reason: It looks like $ will cause an error, but it won’t. In java, identifier rule says, identifier can start with any alphabet or underscore (“_”) or dollar (“$”). So answer is Nothing.

6. Answer: a) Nothing
   Reason: We can overload main() too. But JVM will always call main() that has String[] argument.

7. Ans. (d)
   Explanation: Final and static methods cannot be overridden.

8. Ans. (c)
   Explanation: The overriding method must have same signature, which includes, the argument list and the return type.

9. Ans: (d)
   Explanation: The overriding method can not have more restrictive access modifier.

10. Ans. (a)
    Explanation: super keyword is used to invoke the overridden method from a child class explicitly.

11. Output: Compiler Error
    In the above program, there are no access permission issues because the Point and Main are in the same package and protected members of a class can be accessed in other classes of the same package. The problem with the code is: there is no default constructor in Point.
    So there are following two changes to Point class that can fix the above program:
    1. Remove the parameterized constructor.
    2. Add a constructor without any parameter.

12. Output: String
    Explanation : In case of method overloading, the most specific method is chosen at compile time. As ‘java.lang.String’ is a more specific type than ‘java.lang.Object’. In this case the method which takes ‘String’ as a parameter is choosen.

13. Output: Compile Error at line 19.
    Explanation: In this case of method Overloading, the most specific method is choosen at compile time.
    As ‘java.lang.String’ and ‘java.lang.Integer’ is a more specific type than ‘java.lang.Object’,but between ‘java.lang.String’ and ‘java.lang.Integer’ none is more specific. In this case the Java is unable to decide which method to call.

14. Output: abcd abc false
    Explanation : In Java, String is immutable and string buffer is mutable. So string s2 and s1 both pointing to the same string abc. And, after making the changes the string s1 points to abcd and s2 points to abc, hence false.

15. Output:
    a
    b
    c
    Explanation: While creating a new object of ‘Third’ type, before calling the default constructor of Third class, the default constructor of super class is called i.e, Second class and then again before the default constructor of super class, default constructor of First class is called. And hence gives such output.

16. Answer : B) Writing book
    Explanation : Since static methods can't be overridden, it doesn't matter which class object is created. As a is a Author referenced type, so always Author class method is called. If we remove write() method from Author class then Writer class method is called, as Author class extends Writer class.

17. Output: Compilation fails.
    Explanation: The method calc() in class superClass is final and so cannot be overridden.

18. Answer (c)
    Explanation: When overriding a method of superclass, the method declaration in subclass cannot be more restrictive than that declared in the superclass.

19. Ans. (d)
    Explanation: Once an exception occurs in try block, the execution passes to corresponding catch statement and doesn't return back to try block. Only one of the catch blocks are executed at a time. finally block is always executed whether or not the exception occurred.

20. Ans. (a)
    Explanation: If an exception is handled in the catch statement, the program continues with its normal execution, after executing the catch statement corresponding to that exception. Also, when an exception occurs in the try block, the rest of the program in the try block is not executed.

21. Ans. (a)
    Explanation: The only case when the code inside finally block is not executed is when System.exit(0) is called explicitly in the program. Then exit statement is called and the program is terminated without executing any further.

22. Ans. (a)
    Explanation: A private constructor cannot be used to initialize an object outside the class that it is defined within because it is no longer visible to the external class.

23. Ans. (c)
    Explanation: static variables are associated with the class and are therefore, not allowed inside a method body.

24. Ans. (b)
    Explanation: && operator is evaluated from left to right. If the first expression of && operator evaluates to false, then the second operator is not evaluated. There is no compilation error because divide by 0 is a runtime exception.

25. Ans. (d)
    Explanation: Converting from int to String as well as converting from String to int is not allowed in java.

26. Output: fa la
    Explanation: B b = new A(); b is object of type B, and hence b.sing() refers to the method sing of class B

27. Output: Compiler Error
    Explanation: Final methods cannot be overridden. For Detail see final keyword. However, if we remove the keyword final then, the output will be: called from Derived

28. Output: D. No output : Blank Screen
    Explanation: There are no constructor in this program . This is a method just like name of class, but it is not constructor because constructor does not be return type. So in this program output is blank screen.

29. Output: C. Compile time error
    Explanation : In java, memory of object is created by ”new” keyword. Here only reference variable is created. And reference variable is uninitialized, so reference variable is blank. And we can’t use the blank variable without initialize. So, it gives the compile time error.

30. The answer is option (2)
    Explanation : In the condition of if statement, we assigning are false to b which return a boolean value which is false. Therefore, the control goes to the else part and the output is BYE.

31. The answer is option (4)
    Explanation: In the above for loop it will go for infinite loop and the above program does not give any chance to the next lines of the program. That’s why compiler will give compile time error saying error: unreachable statement.

32. Output: The answer is option (2).
    Explanation: The argument of if statement should be Boolean type. By mistake if we are trying to provide any other data type then we will get compile-time error saying incompatible types. Here the argument is of int type, therefore we will get compile time error saying error: incompatible types: int cannot be converted to Boolean

33. Output: The answer is option (4)
    Explanation: Curly braces are optional in if part. Without curly braces only one statement is allowed under if. If we will try to give more than one statement then we will get compile time error saying error: 'else' without 'if'.

34. Output: The answer is option (4).
    Explanation: In the above program, there is a declarative statement in the if statement without curly braces that's why we will get compile time error saying Error: variable declaration not allowed here.

35. Output: The answer is option (1)
    Explanation: ;(semicolon) is a valid java statement which is also known as empty statement. Therefore we can apply it in if statement also.

36. Output: The answer is option (4)
    Explanation: Every case label should be constant otherwise we will get compile time error. But we can add variable as case label but we have to declare that variable as final. But here we are using variable y as a case label that's why we will get compile time error saying error: constant expression required.

37. Output: The answer is option (3)
    Explanation: Every case label should be constant otherwise we will get compile time error. But we can add variable as case label but we have to declare that variable as final. But here we are using variable y as a case label that's why we will get result as HELLO.

38. Correct Answer: A
    Runtime error because main method is not static.

39. Correct Answer: B
    `x.concat(y);` will create a new string but it's not assigned to x, so the value of x is not changed.

40. Correct Answer: B
    The given statements output will be "false" because in java + operator precedence is more than == operator. So the given expression will be evaluated to
    "s1 == s2 is:abc" == "abc" i.e false.

41. Answer: Option C
    Explanation: Line 1 creates two, one referred to by x and the lost String "xyz". Line 2 creates one (for a total of three). Line 3 creates one more (for a total of four), the concatenated String referred to by x with a value of "xyzabc".

42. The output of the above code will be:

    30Javatpoint

    Javatpoint1020

    Explanation: In the first case, 10 and 20 are treated as numbers and added to be 30. Now, their sum 30 is treated as the string and concatenated with the string Javatpoint. Therefore, the output will be 30Javatpoint. In the second case, the string Javatpoint is concatenated with 10 to be the string Javatpoint10 which will then be concatenated with 20 to be Javatpoint1020.