Dashboard Courses PW Skills Lab Job Portal Experience Portal Become an affiliate Hall of Fame RAHUL

Java Milestone 2nd Exam quiz

59 out of 60 correct

1. Which of the following is not an OOPS concept in Java?

○ Inheritance

○ Encapsulation

○ Polymorphism

● Compilation

Explanation: There are 4 OOPS concepts in Java. Inheritance, Encapsulation, Polymorphism and Abstraction.

2. Which of the following is a type of polymorphism in Java?

● Compile time polymorphism

○ Execution time polymorphism

○ Multiple polymorphism

○ Multilevel polymorphism

Explanation: There are two types of polymorphism in Java. Compile time polymorphism (overloading) and runtime polymorphism (overriding).

3. When does method overloading is determined?

○ At run time

● At compile time

○ At coding time

○ At execution time

Explanation: Overloading is determined at compile time. Hence, it is also known as compile time polymorphism.

4. When Overloading does not occur?

○ More than one method with same name but different method signature and different number or type of parameters

○ More than one method with same name, same signature but different number of signature

○ More than one method with same name, same signature, same number of parameters but different type

● More than one method with same name, same number of parameters and type but different signature

Explanation: Overloading occurs when more than one method with same name but different constructor and also when same signature but different number of parameters and/or parameter type.

5. Which concept of Java is a way of converting real world objects in terms of class?

○ Polymorphism

○ Encapsulation

● Abstraction

○ Inheritance

Explanation: Abstraction is the concept of defining real world objects in terms of classes or interfaces.

6. Which concept of Java is achieved by combining methods and attributes into a class?

● Encapsulation

○ Inheritance

○ Polymorphism

○ Abstraction

Explanation: Encapsulation is implemented by combining methods and attributes into a class. The class acts like a container of encapsulating properties.

7. Which of these keywords is used to define interfaces in Java?

● interface

○ Interface

○ intf

○ Intf

8. Which of these can be used to fully abstract a class from its implementation?

○ Objects

○ Packages

● Interfaces

○ None of the Mentioned

9. Which of these access specifiers can be used for an interface?

● public

○ protected

○ private

○ All of the mentioned

Explanation: Access specifier of an interface is either public or no specifier. When no access specifier is used then default access specifier is used due to which interface is available only to other members of the package in which it is declared, when declared public it can be used by any code.

10. What is the return type of lambda expression?

○ String

○ Object

○ void

● Function

Explanation: Lambda expression enables us to pass functionality as an argument to another method, such as what action should be taken when someone clicks a button.

11. Which among the following can show polymorphism?

○ Overloading ||

○ Overloading +=

● Overloading <<

○ Overloading &&

Explanation: Only insertion operator can be overloaded among all the given options. And the polymorphism can be illustrated here only if any of these is applicable or being overloaded. Overloading is a type of polymorphism.

12. Which among the following can't be used for polymorphism?

● Static member functions

○ Member functions overloading

○ Predefined operator overloading

○ Constructor overloading

Explanation: Static member functions are not property of any object. Hence it can't be considered for overloading/overriding. For polymorphism, function must be property of the object, not only of class.

13. How many basic types of inheritance are provided as OOPs feature?

● 4

○ 3

○ 2

○ 1

Explanation: There are basically 4 types of inheritance provided in OOP, namely, single level, multilevel, multiple and hierarchical inheritance. We can add one more type as Hybrid inheritance but that is actually the combination of any types of inheritance from the 4 basic ones.

14. Which of these keywords is used by a class to use an interface defined previously?

○ import

○ Import

● implements

○ Implements

Explanation: interface is inherited by a class using implements.

15. Which of the following is the correct way of implementing an interface salary by class manager?

○ class manager extends salary {}

● class manager implements salary {}

○ class manager imports salary {}

○ none of the mentioned

16. Which of the following are valid lambda expressions?

○ String a, String b -> System.out.print(a+ b);

○ () -> return;

● (int i) -> i;

○ (int i) -> i++; return i;

Explanation: The body doesn't need to use the return keyword if it only has one statement.

17. Given below code snippet

```
interface A {
    int aMethod(String s);
}
```

○ A a = a -> a.length();

○ A x = y -> {return y;};

○ A s = "2" -> Integer.parseInt(s);

● A b = (String s) -> 1;

Explanation: because it takes a String argument and returns an int value.

18. A lambda expression can be used...

○ As a method argument

○ As a conditional expression in an if statement

○ In a return statement

● Both A and C

19. Which of the following statements are true?

● Curly brackets are required whenever the return keyword is used in a lambda expression

○ A return keyword is always required in a lambda expression

○ A return keyword is always optional in a lambda expression

○ Lambda expressions don't return values

Explanation: A return keyword is not always required (or optional) in a lambda expression. It depends on the signature of the functional interface method. Curly brackets are required whenever the return keyword is used in a lambda expression. Both can be omitted if the lambda expression's body is just one statement.

20. When does Exceptions in Java arise in code sequence?

● Run Time

○ Compilation Time

○ Can Occur Any Time

○ None of the mentioned

Explanation: Exceptions in Java are run-time errors.

21. Which of these keywords is not a part of exception handling?

○ try

○ finally

● thrown

○ catch

Explanation: Exceptional handling is managed via 5 keywords – try, catch, throws, throw and finally

22. Which of these keywords must be used to handle the exception thrown by try block in some rational manner?

○ try

○ finally

○ throw

● catch

Explanation: If an exception occurs within the try block, it is thrown and cached by the catch block for processing.

23. Which of these keywords is used to manually throw an exception?

○ try

○ finally

● throw

○ catch

24. The class at the top of the exception class hierarchy is ....

○ ArithmeticException

● Throwable

○ Object

○ Exception

25. Which keyword is used to explicitly throw an exception?

○ try

○ throwing

○ catch

● throw

26. In which of the following package Exception class exist?

○ java.util

○ java.file

○ java.io

● java.lang

○ java.net

27. Exceptions generated in try block are caught in .... block.

● catch

○ throw

○ throws

○ finally

28. What will be the output of the following Java program?

```
class Launch
{
    public static void main(String args[])
    {
        try
        {
            System.out.print("Hello" + " " + 1 / 0);
        }
        catch(ArithmeticException e)
         {
    System.out.print("World");
        }
    }
}
```

○ Hello

● World

○ HelloWorld

○ Hello World

Explanation: System.out.print() function first converts the whole parameters into a string and then prints, before “Hello” goes to output stream 1 / 0 error is encountered which is cached by catch block printing just “World”.

29. What will be the output of the following Java program?

```
class Demo
{
   public static void main(String args[])
   {
     try
     {
         int a, b;
         b = 0;
         a = 5 / b;
        System.out.print("A");
      }
      catch(ArithmeticException e)
      {
   System.out.print("B");
      }
   }
}
```

○ A

● B

○ Compilation Error

○ Runtime Error

30. Members which are not intended to be inherited are declared as___

○ Public members

○ Protected members

● Private members

○ Private or Protected members

Explanation: Private access specifier is the most secure access mode. It doesn't allow members to be inherited. Even Private inheritance can only inherit protected and public members.

31. If a derived class object is created, which constructor is called first?

● Base class constructor

○ Derived class constructor

○ Depends on how we call the object

○ Not possible

Explanation: First the base class constructor is invoked. When we create a derived class object, the system tries to invoke its constructor but the class is derived so first the base class must be initialized, hence in turn the base class constructor is invoked before the derived class constructor.

32. The private members of the base class are visible in derived class but are not accessible directly.

● True

○ False

33. How can you make the private members inheritable?

○ By making their visibility mode as public only

○ By making their visibility mode as protected only

○ By making their visibility mode as private in derived class

● It can be done both by making the visibility mode public or protected

Explanation: It is not mandatory that you have to make the visibility mode either public or protected. You can do either of those. That will give you permission to inherit the private members of the base class.

34. Which exception is thrown when java is out of memory?

○ MemoryError

● OutOfMemoryError

○ MemoryOutOfBoundsException

○ MemoryFullException

Explanation: The Xms flag has no default value, and Xmx typically has a default value of 256MB. A common use for these flags is when you encounter a java.lang.OutOfMemoryError

35. Which one of the following is not an access modifier?

○ protected

● void

○ public

○ private

Explanation: Public, private, protected and default are the access modifiers.

36. What is multithreaded programming?

○ It's a process in which two different processes run simultaneously

● It's a process in which two or more parts of same process run simultaneously

○ It's a process in which many different process are able to access same information

○ It's a process in which a single process can access information from many sources

37. Which of these are types of multitasking?

○ Process based

○ Thread based

● Process and Thread based

○ None of the mentioned

Explanation: There are two types of multitasking: Process based multitasking and Thread based multitasking.

38. Thread priority in Java is?

● Integer

○ Float

○ double

○ Long

Explanation: Java assigns to each thread a priority that determines how that thread should be treated with respect to others. Thread priority is integers that specify relative priority of one thread to another.

39. What requires less resources?

● Thread

○ Process

○ Thread and Process

○ Neither Thread nor Process

Explanation: Thread is a lightweight and requires less resources to create and exist in the process. Thread shares the process resources.

40. What does not prevent JVM from terminating?

○ Process

● Daemon Thread

○ User Thread

○ JVM Thread

Explanation: Daemon thread runs in the background and does not prevent JVM from terminating. Child of the daemon thread is also a daemon thread.

41. What decides thread priority?

○ Process

○ Process scheduler

○ Thread

● Thread scheduler

Explanation: Thread scheduler decides the priority of the thread execution. This cannot guarantee that higher priority thread will be executed first, it depends on thread scheduler implementation that is OS dependent.

42. Encapsulation is supported by ___________?

○ Object.

○ Method.

● Classes.

○ None of the above.

43. Which of these classes are the direct subclasses of the Throwable class?

○ RuntimeException and Error class

○ Exception and VirtualMachineError class

● Error and Exception class

○ IOException and VirtualMachineError class

44. Types of exceptions in Java programming are

○ Checked exception

○ unchecked exception

● Both A & B

○ None

45. Checked exception caught at

● compile time

○ run time

○ Both at compile and run time

○ None

46. What exception can occur in the below java program if we access 5 elements in the array that does not exist?

```
public class TException {


public static void main(String[] args) {
try {
int a[] = { 5, 10, 15, 20 };
System.out.println("Element :" + a[4]);
}
finally{}
}
}
```

● ArrayIndexOutOfBoundsException

○ ArithmeticException

○ NullPointerException

○ None

Explanation: ArrayIndexOutOfBoundsException unchecked exception will occur at run time when we execute the program.

47. Which is the super class of all java exceptions classes?

○ Exception

○ RuntimeException

● Throwable

○ IOException

48. Which one is the correct declaration for implementing two interfaces?

Consider interface A and B. Class C wants to implement both interfaces

● class C implements A, B

○ class C implements A, implements B

○ class C implements A extends B

49. The 'implements' keyword is used to ___.

○ Implement the function of a class

● Inherit an interface in Java

○ Inherit a class in java

○ All of these

50. What will be the output of the following Java program?

```
public class ExceptionTest
{
 public static void main(String[] args)
 {
    try
    {
      int a=10/1;
      try
      {
int b=20/1;
}
      catch(Exception e1)
      {
System.out.println("b=20"); }
    }
    catch(Exception e2)
    {
System.out.println("a=10");}
  }
}
```

○ a=10
b=20

○ b=20
a=10

○ Compiler error

● No output

51. Encapsulation is also called as?

● Data Hiding.

○ Data Hidding.

○ Data Encapsulation.

○ None of the above.

52. The variables initialized inside a TRY-with-resource are treated like _ _ _ _ variables in Java.

○ static

○ instance

● final

○ None of the above

53. Which of the following is not an advantage to using inheritance?

○ Code that is shared between classes needs to be written only once.

○ Similar classes can be made to behave consistently.

○ Enhancement to the base class will automatically be applied to the derived classes.

● One big super class can be used instead of many little classes.

54. What will be the output of the following Java program?

```
class TestA {
public void start() {
System.out.println("TestA"); }
}
public class TestB extends TestA {
public void start() {
System.out.println("TestB"); }//overridden method
public static void main(String[] args) {
((TestA)new TestB()).start();
}
}
```

What is the result?

○ TestA

● TestB

○ Compilation fails.

○ An exception is thrown at runtime.

55. Analyze the following code:

```
public abstract class Test implements Runnable{
public void doSomething(){};
}
```

○ The program will not compile because it does not implement the run() method.

○ The program will not compile because it does not contain abstract methods.

● The program compiles fine.

○ None of the above

56. What notifyAll() method do?

○ Wakes up one threads that are waiting on this object's monitor

○ Wakes up all threads that are not waiting on this object's monitor

● Wakes up all threads that are waiting on this object's monitor

○ None of the above

57. 'catch' block gets skipped if no exception occurred in the 'try' block

● True

○ False

58. 'throw' keyword is used to _______

○ Propagate the exception to the caller of the method

○ Define a new exception type

● Throw an exception explicitly

○ None of the above

59. Predict the output of following Java program

```
class Main {
  public static void main(String args[]) {
    try {
      throw 10;
    }
    catch(int e) {
      System.out.println("Got the  Exception " + e);
    }
  }
}
```

○ Got the Exception 10

○ Got the Exception 0

● Compile Error

○ Compiles and runs fine

60. Output of following Java program?

```
class Main {
  public static void main(String args[]) {
    int x = 0;
    int y = 10;
    int z = y/x;
  }
}
```

○ Compiler Error

○ Compiles and runs fine

● Compiles fine but throws ArithmeticException exception

○ No Output

Submit