

JAVA

Contents

AWT.....	2
CLONE REFLECTION API.....	5
COLLECTION API.....	6
DESIGN PATTERN	9
EXCEPTION	10
FILE HANDLING.....	19
FUNDAMENTAL.....	24
GENERIC	29
HIBERNATE	30
INHERITANCE	35
JAVA FX.....	48
JDBC.....	48
JSF.....	52
JSP	52
MULTITHREADING.....	60
MVC.....	68
Oops	68
SERVLET.....	80
SOCKET PROGRAMMING	89
STRUTS	92
SPRING	95
SWING.....	97

AWT

1. Adapter class is not available for

a: ItemListener

b: MouseListener

c: KeyListener

d: WindowListener

2. Given

```
public class MyApp extends Applet
```

```
{  
    public MyApp(int k)  
    {  
    }  
}
```

What will happen to the above code?

a: compilation error "cannot instantiate MyApp"

b: runtime error "paint() method not available"

c: runtime error "InstantiationException"

d: compilation error "paint() not defined"

3. Given

```
public class MyApp2 extends Applet
```

```
{  
    @Override  
    public void init()  
    {  
        setLayout(new GridBagLayout());  
        GridBagConstraints gbc=new GridBagConstraints();  
        gbc.gridwidth=3;  
        gbc.gridheight=2;  
        add(new Button("ok"));  
    }  
}
```

What will happen?

a: compiler error "add method must take 2nd argument as GridBagConstraints"

b: exception during runtime

c: Button will appear according to gridwidth and gridheight specified

d: Button will appear but not according to gridwidth and gridheight specified.

4. Select correct statement from the following

a: BorderLayout is the default layout for Applet

b: GridLayout can not work without GridBagConstraints

c: pack() method displays window in a preferred size

d: FlowLayout can not be used for swing components

5. Given

```
setLayout(new BorderLayout());
```

```
add("south",new TextField(20));
```

What will happen to the above code?

- a: compiler error c: textfield will be displayed properly at south
b: exception d: textfield will be displayed in the center,since u have given illegal argument.

6. Select the wrong statements from the following

- a: Applet extends Panel c: **Dialog extends Frame**
b: FileDialog extends Dialog d: Window extends Container

7. Given

```
public class Trial extends Frame
```

```
{  
    public Trial(String mess)  
    {  
        MenuBar mb=new MenuBar();  
        // here  
    }  
}
```

How will u add "mb" to the frame?

- a: addMenuBar(mb); **b: setMenuBar(mb);** c: mb.addMenuBar(); d: add(mb);

10. java.awt.Component class method getLocation() returns Point (containg x and y cordinate).What does this x and y specify

- a: Specify the position of components lower-left component in the coordinate space of the component's parent.
b: Specify the position of components upper-left component in the coordinate space of the component's parent.
c: Specify the position of components upper-left component in the coordinate space of the screen.
d: None of the above

11. When u invoke "repaint()", for a lightweight component , the AWT package calls which component method?

- a) repaint() b) **update()** c) paint() d) draw()

12. What does the following line of code do?

```
TextField tf=new TextField(30);
```

- a) This code is illegal , as there is no such constructor available inside "TextField" class.
b) Creates the TextField object, that can hold 30 rows, but since it is not initialized to anything, it will be always empty.
c) Creates a new TextField object that is 30 columns of text.
d) This code creates a TextField object that can hold 30 rows of text

13. Which of the following the valid way to embed an applet class named myapplet into a web page.
- a) `<applet class=myapplet.class width=100 height=100> </applet>`
 - b) `<applet code=myapplet width=100 height=100> </applet>`
 - c) `<applet code=myapplet.class height=100 width=100 > </applet>`**
 - d) `<applet param=myapplet.class width=100 height=100> </applet>`
14. What is the purpose of “code” attribute of the applet tag?
- a) A URL that points to the class of the applet.**
 - b) A URL to the applet when it is stored in jar or zip file.
 - c) Indicate the base URL of the applet if the code attribute is relative.
 - d) Defines the horizontal spacing around the applet.
15. Executable applet is nothing but _____ file of applet.
- a) class b) java c) html d) applet
16. Select correct statement from the following
- a) Invisible components are required in SwingLayout
 - b) BorderLayout is the default layout for JApplet**
 - c) the default lookandfeel for swing components is MotifLookAndFeel.
 - d) swing does not have DelegationEvent model.
17. Method to apply menubar to the swing container is:
- a) addMenuBar()
 - b) setJMenuBar()**
 - c) setSMenuBar()
 - d) setMenuBar()
18. Select wrong statement from the following
- a) FlowLayout is the default layout for Applet.
 - b) By default Frame is invisible.
 - c) pack() method displays window in a preferred size
 - d) None of these.**
19. Given
- ```
setLayout(new BorderLayout());
add(new TextField(20));
```
- What will happen to the above code ?
- a) compiler error
  - b) exception
  - c) textfield will not be displayed since u haven't mentioned an area.

d) textfield will be displayed in the center.

20. Given

```
import java.awt.*;
public class MyFr2
{
 Button b1,b2;
 public MyFr2(String title)
 {
 Frame f=new Frame(title);
 f.setLayout(new BorderLayout());
 b1=new Button("ok");
 b2=new Button("cancel");
 f.setLayout(new FlowLayout());
 f.add(b1);
 f.add(b2);
 f.setSize(400,400);
 f.setVisible(true);
 }
 public static void main(String args[])
 {
 new MyFr2("My Window");
 }
}
```

What will happen to the above code ?

- a) compiler error "can not set layout twice"
- b) frame will be displayed with only one "cancel" button
- c) **frame will be displayed with two buttons.**
- d) exception during runtime.

21. Which method is required to read parameters pass to Applet?

- a) **getParameter**
- b) getInitParameter
- c) getAppletParameter
- d) none of these

22. What is sent to the user via HTTP, invoked using the HTTP protocol on the user's computer and run on the user's computer as an application?

- a) A Java application
- b) A Java Servlet
- c) **A Java applet**
- d) None of the above

### CLONE REFLECTION API

- 1) Cloneable interface contains "clone()" method
  - A. True
  - B. False

- 2) Clone method is declared as throws
- A. IOException
  - B. CloneNotFoundException
  - C. CloneNotSupportedException**
  - D. None of the above
- 3) Clone() method in Object class is
- A. **Protected**
  - B. Public
  - C. Default
  - D. Private
- 4) If u override "clone()" method u can apply access modifier
- A. Protected
  - B. Public
  - C. protected or public**
  - D. Default
- 5) By default "clone" method does
- A. Shallow copy**
  - B. Deep copy
  - C. Shallow and deep both copies
  - D. None
- 6) Interface which does not contain any method is called as
- A. Empty
  - B. Methodless
  - C. Marker**
  - D. Void
- 7) Inner class methods can access outer class members directly
- A. **True**
  - B. False
- 8) Static nested class methods can access outer class members directly
- A. True**
  - B. **False**
- 9) There is one instance of class "Class" per class loaded.
- A. True**
  - B. False
- 10) To instantiate a particular class through reflection api we use
- A. New Class
  - B. Class.newInstance**
  - C. Class.newCreate
  - D. None of the above

### COLLECTION API

1. One of the following throws ConcurrentModificationException if we try to modify while iterating over it.
- A: Hashtable
  - B: CopyOnWriteArrayList

### C: ArrayList

D: ConcurrentHashMap

2. The default capacity and load factor for Map implementations are

A: 12 and 0.60

**B: 16 and 0.75**

C: 20 and 0.75

D: 18 and 0.60

3. Given

```
Class Animal{void eat(){}}
```

```
Class Dog extends Animal{}
```

```
Class Cat extends Animal{}
```

```
Void disp(List<? super Dog> mylist)
```

Which of the following is the wrong argument to disp ?

A: ArrayList of Animal

B: ArrayList of Dog

C: ArrayList of Object

**D: All the above are correct arguments.**

4. Which statement is true ?

A: List<?> will allow u to add inside list.

**B: List<Object> will allow u to add inside list**

C: both A and B

D: we can pass ArrayList<Integer> to List<Object>

5. Which collection class allows you to grow or shrink its size and provides indexed access to its elements, but whose methods are not synchronized?

A: java.util.HashSet

B: java.util.LinkedHashSet

C: java.util.List

**D: java.util.ArrayList**

6. Which of the following class uses String as key to store the value in object?

a) Dictionary

b) Array

c) ArrayList

**d) Properties**

7. Which of these class objects uses key to store value?

a) Hashtable

b) Dictionary

c) Map

**d) all if the mentioned**

8. \_\_\_\_\_ can be used to control the order of certain data structure and collection of object too.

- a) Serial comparators      b) natural comparators      c) **comparators**      d) all of the above

9. How does the set collection deal with duplicate elements?

- A. An exception is thrown if you attempt to add an element with a duplicate value  
**B. The add method returns false if you attempt to add an element with a duplicate value**  
C. A set may contain elements that return duplicate values from a call to the equals method  
D. Duplicate values will cause an error at compile time

10. What is the sequence followed by HashMap or HashSet while adding or retrieving entries.

- A: ==, equals(), hashCode()  
B: equals(), == , hashCode()  
**C: hashCode() , == , equals()**  
D: none of these

11. If you try to invoke “remove()” method on iterator of CopyOnWriteArrayList , it raises following exception

- A: ConcurrentModificationException  
**B: UnsupportedOperationException**  
C: IllegalStateException  
D: none of these

12. Map implementation which provides both Thread-Safety as well as Concurrency.

- A: ConcurrentHashMap**  
B: HashMap  
C: Hashtable  
D: none of these

13. Stream API is used to implement

- A. Internal iteration**  
B. External iteration  
C. Both A and B  
D. None of the above

14. In get () or put() of map implementation equals () is Called before ==.

- A. True      **B. False**

15. Algorithms are present inside.

- A. LinkedList  
B. Collection  
**C. Collections**  
D. Hashtable

16. Iterator of ArrayList is Fail-Safe.



A. False

B. True

17. All the Collection API implementation classes implement\_\_\_\_\_.

A. Runnable

**B. Serializable**

C. Externalizable

D. Comparable

18. When you add any object inside Collection API implementation class, its copy is added.

A. True

**B. False**

19. Whenever we create any implementation of set it result into\_\_\_\_\_.

A. Vector

B. None of these

C. List

**D. Map**

20. In map implementation when hashCode of two keys are same it is called as?

A. Hashing

**B. Hash Collision**

C. Hash Clash

D. None of these

21. One of the following allows us to define more than one strategies.

**A. Comparator**

B. None of these

C. Enumeration

D. Comparable

22. Snapshot of list is created in case of \_\_\_\_\_.

**A. CopyOnWriteArrayList**

B. Linked List

C. ArrayList

D. Vector

23. One of the followings is not Thread Safe

A. StringBuffer

B. Hashtable

C. Vector

**D. none of these**

## DESIGN PATTERN

1. Which design pattern would you use to have a prototypical instance determine the concrete class of object being created?

A: Prototype factory design pattern

B: Virtual prototype design pattern

C: Abstract prototype design pattern

### D: Prototype design pattern

2. Which design pattern is used in the Java Database connectivity JDBC(TM)?

A: Builder design pattern

B: Factory method design pattern

**C: Abstract Factory design Pattern**

D: Singleton design Pattern

3. java.net.InetAddress is a good example of which design pattern?

**A: Factory pattern**

B: Builder pattern

C: Adapter pattern

D: Bridge pattern

4. Which of the following is an example of creational design pattern?

A: Adapter

B: Composite

**C: Builder**

D: Command

5. Which of the following is NOT an advantage of using the Session Facade pattern?

A: Hides model complexity from the client

B: Reduces network traffic

C: Provides a simple interface to the client

**D: Enables the client to control transactions**

6. You are building the server side of an application and you are finalizing the interfaces that you will provide to the presentation layer. However, you have not yet finalized the access details of the business services. Which design pattern should you use to mitigate this concern?

A: Model-View-Controller

B: Data Access Object

**C: Business Delegate**

D: Facade

## EXCEPTION

1. Given Following code:

```
import java.io.*;
class sub extends base
{
 void disp()throws IOException
 {
 }
}
```

```
}
class base
{
 void disp()throws Exception
 {
 }
}
public class myclass
{
 public static void main(String args[])
 {
 try
 {
 base b=new sub();
 b.disp();
 }
 catch(Exception ee)
 {
 System.out.println(ee);
 }
 System.out.println("done");
 }
}
```

- A: warning
- B: compilation error
- C: runtime error

**D: output "done"**

2. Which statement is false from the following?

- A: we can have try and finally without catch
- B: finally gets executed irrespective whether exception is raised or not
- C: if system.exit is called from within try or catch, finally will not be executed at all

**D: none of the above**

3. Class.forName requires which of the following exception to be handled

- A: ClassCastException
- B: ClassNotFoundException**
- C: IllegalAccessException
- D: none of the above

4. Class.newInstance() requires which of the following exception to be handled

- A: IOException
- B: ClassNotFoundException
- C: IllegalAccessException**
- D: none of the above

5. Imagine there are two exception classes Exception1 and Exception2 derived from the Exception class.

Given these two definitions:

```
class First
{
 void test()throws Exception1,Exception1
 {
 }
}
class Second extends First
{
 void test()
 {
 }
}
```

Now define a class "Third" derived from "Second" and override "test ()" method inside it.

What exceptions can Third's test() method throw?

- a) Exception1
- b) Exception2
- c) **No checked exceptions**
- d) it can declare any checked

6. What letters get written to the standard output with the following code?

```
public class MyClass
{
 public static void main(String args[])
 {
 try
 {
 method();
 }
 catch(Exception ie)
 {
 }
 }
 static void method()
 {
 try
 {
 wrench();
 System.out.println("a");
 }
 catch(ArithmeticException ae)
 {
 System.out.println("b");
 }
 finally
 {
 }
 }
}
```

```
 System.out.println("c");
 }
 System.out.println("d");
}
static void wrench()
{
 throw new NullPointerException();
}
}
```

- a) A                      b) b                      c) c                      d) compilation error

7. Which statement is false from the following?

- a. The exceptions that are checked at compilation-time by the Java Compiler are called
- b. 'Checked exception'.
- c. The exceptions that are checked by the JVM are called 'unchecked exception'
- d. Both 1 and 2
- e. **None of the above**

8. Read the following code below.

```
public interface AQuestion
{
 public abstract void someMethod() throws Exception;
}
```

A Class implementing this interface should

- a. Necessarily be an abstract class
- b. Should have the method public abstract void someMethod();
- c. Should have the method public void someMethod() which has to throw an exception which is a subclass of java.lang.Exception.
- d. **Should have the method public void someMethod() which need not throw an Exception.**

9. Given:

```
public class Test
{
 public static void throwIt()
 {
 throw new Exception();
 }
 public static void main(String[] args)
 {
 try
 {
 System.out.println("Hey There");
 }
 finally
 {
 System.out.println("in Finally");
 }
 }
}
```

```
}
}
```

What will happen when one tries to compile and run above code?

- a. **Compilation Fails**
- b. The program will print Hey There, then will print in finally.
- c. The program will print Hey There, then will print that an Exception has occurred, and then will print in finally.
- d. None of them

10 Given:

```
1. public class Foo {
2. public static void main(String[] args) {
3. try {
4. return;
5. } finally {
6. System.out.println("Finally");
7. }
8. }
9. }
```

What is the result?

- a. **Finally**
- b. Blank
- c. Null
- d. None of the above

11. In exception handling mechanism, finally block is always executed, even if no exception occurred in the try block

- a. **True**
- b. False

12. Exceptions can be caught or rethrown to a calling method.

- a. **True**
- b. False

13. Given Following code:

```
import java.io.*;
class base
{
 void disp()throws IOException
 {
 }
}
class sub extends base
{
 void disp()throws Exception
 {
 }
}
```

public class myclass

```
{
 public static void main(String args[])
 {

 }
}
```

- a) compile error                                      b) neither compilation nor runtime error  
c) no compilation error but exception at runtime.

14. What will happen to the following code?

```
public class Test
```

```
{
 public static void aMethod() throws Exception
 {
 try /* Line 5 */
 {
 throw new Exception(); /* Line 7 */
 }
 finally /* Line 9 */
 {
 System.out.print("finally "); /* Line 11 */
 }
 }

 public static void main(String args[])
 {
 try
 {
 aMethod();
 }
 catch (Exception e) /* Line 20 */
 {
 System.out.print("exception ");
 }
 System.out.print("finished"); /* Line 24 */
 }
}
```

- A: finally  
B: exception finished  
**C: finally exception finished**  
D: compilation fails

15. Which statement is true,if the following program is run by `java test10` ?

```
public class test10
{
 public static void main(String []args)
 {
 String []num={"one","two","three","four"};
```

```
 if(args.length==0)
 {
 System.out.println("Zero");
 }
 else
 {
 System.out.println(num[args.length]+" arguments");
 }
 }
}
```

- A. The program won't run because argument of main is not properly mentioned
- B. The program will throw a NullPointerException
- C. The program will display Zero when executed**
- D. The program will display 0 arguments when executed

16. following program will not print "=="

```
public class test12
{
 public static void main(String args[])
 {
 String first="abc";
 String second=new String(first);
 if(first==second)
 {
 System.out.println("==");
 }
 }
}
```

- A. True**
- B.False

17. Assuming a method contains code which may raise an Exception (but not a RuntimeException), what is the correct way for a method to indicate that it expects the caller to handle that exception:

- A. throw Exception
- B. throws Exception**
- C. new Exception
- D. Don't need to specify anything

18. What is the result of executing the following code, using the parameters 4 and 0:

```
public void divide(int a, int b)
{
 try
 {
 int c = a / b;
 }
 catch (Exception e)
 {
 System.out.print("Exception ");
 }
 finally
```



```
{
 System.out.println("Finally");
}
```

- A. Prints out: Exception Finally**
- B. Prints out: Finally
- C. Prints out: Exception
- D. No output

19. Given

```
public class MyClass
{
 public static void main(String args[])
 {
 String s1="hello";
 String s2=new String("hello");
 String s3="hello";
 System.out.println(s1==s2);
 System.out.println(s1==s3);
 System.out.println(s1.equals(s2));
 }
}
```

What will be the output ?

- A. true, true, true
- B. true, false, true
- C. false, true, true**
- D. none of the above

20. specify which of the following is true ?

- A. protected members can not be accessed directly in the same package.
- B. Protected member can be accessed with super class reference in different package.
- C. Private member can be accessed by subclass using super keyword.
- D. Constructors are not inherited.**

21. Can you declare method local variable as final and can an abstract class may be final?

- A. Yes, yes
- B. Yes, no**
- C. No, yes
- D. No, no

22. Which of these methods of String class is used to obtain character at specified index?

- A. char()
- B. charOn()
- C. charat()
- D. charAt()**

23. What will happen in the below code snippet:

```
public class MyClass
```

```
{
 int i;
 float f;
 double d;
 boolean bl;
 public static void main(String args[])
 {

 System.out.println("int = "+i);
 System.out.println("float = "+f);
 System.out.println("double = "+d);
 System.out.println("boolean = "+bl);
 }
}
```

A. Int=0  
float=0.0  
double=0.0  
boolean=false

**B. Compilation error: cannot make static reference to the non-static field**

C. Int=0  
float=0.000  
double=0.000  
boolean=false

D. Compilation error: variable may not have been initialized

24. What is legal?

- A. Try{}catch()
- B. Try{}catch()finally{}
- C. Try{}finally{}
- D. All of the above**

25. What will be returned?

Try{return 1;}catch(){return 2;}finally{return 3;}

- A. 3**
- B. 2
- C. 1
- D. Compilation error

26. One of the following is unchecked exception

- A. IOException
- B. ClassNotFoundException
- C. FileNotFoundException
- D. None of the above**

27. Which one is checked exception

- A. ClassCastException

**B. MalformedURLException**

- C. ArrayIndexOutOfBoundsException
- D. None of the above

28. In order to declare exception which keyword is used

A. Throw

**B. Throws**

- C. Throwing
- D. None of the above

29. Class.forName throws

A. ClassCastException

**B. ClassNotFoundException**

- C. NoClassDefFoundException
- D. ClassLoadingException

30. Checked exceptions are automatically propagated to the caller.

A. True

**B. False**

31. Unchecked exceptions are automatically propagated to the caller.

**A. True**

B. False

32. If u want to create checked exception as user defined exception u need to extend

A. RuntimeException

B. Throwable

**C. Exception**

D. Error

33. When u write one try and multiple catch the most specific catch should precede the most generic catch

**A. True**

B. False

## FILE HANDLING

1. One of the following class provides "seek ()" method

A: FileInputStream

B: File

**C: RandomAccessFile**

D: FileReader

2. Given

```
File f=new File("abc.txt");
```

```
FileInputStream fis=new FileInputStream(f);
```

```
byte arr[]=new byte[100];
```

which statement will read content of "abc.txt" into arr.

A: arr=fis.read()

B: f.read(arr)

C: arr=f.read()

**D: fis.read(arr)**

3. Which one is wrong statement?

**A: FileInputStream fis=new FileInputStream(new BufferedInputStream("abc.txt"));**

B: DataOutputStream dis=new DataOutputStream(new FileOutputStream("xyz.txt"));

C: FileOutputStream fos=new FileOutputStream(new File("aaa.txt"));

D: SequenceInputStream ss=new SequenceInputStream(new FileInputStream("a.txt"),new FileInputStream("b.txt"));

4. Given

```
class base
```

```
{
```

```
 int k;
```

```
}
```

```
class sub extends base implements Serializable
```

```
{
```

```
 int j;
```

```
}
```

If we try to serialize instance of sub class,

A: sub as well as base state will be serialized

B: NotSerializableException

**C: only sub instance will be serialized**

D: compiler error "cannot serialized object having non-serializable parent"

5. Classes that do not implement \_\_\_\_\_ interface will not have any of their State serialize or deserialized.

A: List

B: SingleThreadModel

**C: Serializable**

D: Comparable

6. Which one of the following is not from java.io.package

**a. String - correct ans**

b. StringReader

c. Writer

d. File

7. What is the output?

```
public static void main(String[] args) {
```

```
// TODO Auto-generated method stub
int x=0;
int y=10;
do
{
 y--;
 ++x;
}while(x<5);
System.out.println(x+"\t"+y);
}
```

output- 5 5

how does readObject() of ObjectInputStream indicate end of file?

- a. returns null
- b. "" -1
- c. **throws java.io.EOFException - correct ans**
- d. closes automatically

8. What does the following code do?

```
File f=new File("hello.test");
```

```
FileOutputStream fos=new FileOutputStream(f);
```

- a. **Create a file "hello.test" if it does not exists in write mode.**
- b. Open a file named "hello.test", so that u can write to it and read from it but does not create the file if it is not existing yet.
- c. Open a file named "hello.test", so that u can write to it and read from it.
- d. Create an object that you can now use to create and open the file named "hello.test" and write to and read from the file.

9. Given this code:

```
Import java.io.*;
```

```
Class Write
```

```
{
 Public static void main(String args[])
 {
 File f=new File("a.txt");
 FileOutputStream fos=new FileOutputStream(f);
 // write int here inside the file
 }
}
```

How can u replace the comment at the end of main with code that will write integers from 0 to 9 ?

```
a) DataOutputStream dos=new DataOutputStream(fos);
 for(int i=0;i<=9;i++)
 {
 dos.write(i);
 }
```

- b) 

```
for(int i=0;i<=9;i++)
{
 f.writeInf(i);
}
```
- c) 

```
for(int i=0;i<=9;i++)
{
 fos.writeInt(i);
}
```
- d) 

```
DataOutputStream dos=new DataOutputStream(fos);
for(int i=0;i<=9;i++)
{
 dos.writeInt(i);
}
```

10. What is the permanent effect on the file system of writing data to a new `FileWriter("report")`, given the file report already exists?

1. The data is appended to the file
- 2. The file is replaced with a new file**
3. An exception is raised as the file already exists
4. The data is written to random locations within the file

11. Which one is wrong statement?

- A: `FileInputStream fis=new FileInputStream("abc.txt");`  
B: `DataOutputStream dis=new DataOutputStream(new FileOutputStream("xyz.txt"));`  
C: `FileOutputStream fos=new FileOutputStream(new File("aaa.txt"));`  
**D: `FileOutputStream fos=new FileOutputStream(new ObjectOutputStream("aaa.txt"));`**

12. Which statement is correct?

- A: `Externalizable` is a base interface of `Serializable`  
B: `String` class is final hence cannot be serialized  
**C: When a class implements `Serializable` and it is deserialized using `readObject()`, constructor is never invoked.**  
D: `Externalizable` is a marker interface.

13. Given

class base

```
{
 int k;
}
```

class sub implements `Serializable`

```
{
 base b=new base();
 int j;
```

}

If we try to serialize instance of sub class,

A: sub as well as base state will be serialized

**B: NotSerializableException**

C: only sub instance will be serialized

D: compiler error “ cannot serialized object having non-serializable parent”

14. Which class is not serialized

**A: java.lang.Thread**

B: java.lang.Applet

C: java.lang.Class

D: All of the above

15. \_\_\_\_\_ is a communication path bet’n source and destination

A. File

B. **stream**

C. directory

D. none of the above

16. InputStream and OutputStream are concrete classes

A. True

**B. false**

17. if u want to write primitive types u need to use

A. **DataOutputStream**

B. FileOutputStream

C. OutputStream

D. ObjectOutputStream

18. \_\_\_\_\_ class allows us to write and read both.

A. FileReaderWriter

B. **RandomAccessFile**

C. BufferedWriter

D. none of the above

19. Serializable extends Externalizable

A. True

**B. false**

20. Serializable is marker interface.

**A. True**

B. false

21. In case of Serializable when u deserialize an object constructor does not get invoked.

**A. True**

B. false

22. While deserialization if serialVersionUID does not match we get

A. **IllegalClassException**

B. **InvalidClassException**

- C. NullPointerException
- D. none of the above

23. Which is correct

- A. `a)FileOutputStream fos=new FileOutputStream(object to be added);`  
`ObjectOutputStream oos=new ObjectOutputStream("filename");`  
`oos.writeObject();`
- B. `FileOutputStream fos=new FileOutputStream("filename");`  
`ObjectOutputStream oos=new ObjectOutputStream(object to be added);`  
`oos.writeObject();`
- C. **`FileOutputStream fos=new FileOutputStream("filename");`**  
**`ObjectOutputStream oos=new ObjectOutputStream(fos);`**  
**`oos.writeObject(object to be added);`**
- D. none of the above

24. File class is used to create new file.

- A. True
- B. **false**

25. in case of Externalizable when u deserialize an object first `readExternal()` is called and then constructor is called.

- A. True
- B. **false**

26. In order to serialize inner class, outer class must be Serializable

- A. True**
- B. false

27. If inner class implements Externalizable we don't get any problem while deserialization

- A. True
- B. **false**

28. If static nested class implements Externalizable we don't get any problem while deserialization

- A. True**
- B. false

29. `Java.lang.Object` class implements Serializable

- A. True
- B. **false**

### FUNDAMENTAL

1. Given the following code

```
public class test10
{
 public static void main(String args[])
 {
 int x,y;
 y=10;
 }
}
```



```
}
```

On Compilation of above code

A: Local variable y might not have been initialized

**B: no error**

C: Local variable x might not have been initialized

D: Local variable x initialized but not declared.

2. Given the following code

```
public class test10
```

```
{
```

```
 public static void main(String args[])
```

```
 {
```

```
 int x,y;
```

```
 y=10;
```

```
 System.out.println(x+"\t"+y);
```

```
 }
```

```
}
```

On Compilation of above code

A: Local variable y might not have been initialized

B: no error

**C: Local variable x might not have been initialized**

D: Local variable x initialized but not declared.

3. Given the following class, which statements if inserted at position 1 will cause a compilation error?

```
public class test1
```

```
{
```

```
 int a;
```

```
 int b=0;
```

```
 static int c;
```

```
 public void disp()
```

```
 {
```

```
 int d;
```

```
 int e=0;
```

```
 // Position 1
```

```
 }
```

```
}
```

A: a++

B: b++

C: c++

**D: d++**

4. Scanner class is the part of

A: java.io

B: java.scanner

**C: java.util**

D: java.lang

5. What will happen in the below code snippet:

```
public class MyClass
```

```
{
```

```
 int i;
```

```
 float f;
```

```
 double d;
```

```
 boolean bl;
```

```
 public static void main(String args[])
```

```
 {
```

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

```
 System.out.println("float = "+f);
```

```
 System.out.println("double = "+d);
```

```
 System.out.println("boolean = "+bl);
```

```
 }
```

```
}
```

a)     Int=0

float=0.0

double=0.0

boolean=false

b)     Int=0

float=0.000

double=0.000

boolean=false

**c)     Compilation error: cannot make static reference to the non-static field**

d)     Compilation error: variable may not have been initialized

6. public class ForCheck{

```
public static void main(String[] args){
```

```
for(int x=5 ; x>=2 ; --x)
```

```
System.out.print(x);
```

```
}
```

```
}
```

A: **5432**

B: 432

C: 43

D: 543

7. Given:

- 1) class OOP{
- 2) public static void main(String[] args) {
- 3) doStuff(1);
- 4) doStuff(1 , 2);
- 5) }
- 6) // insert code here
- 7) }

Which of the following inserted independently at line 6, will compile?

A: **static void doStuff(int... doArgs) { }**

B: static void doStuff(int[] doArgs) { }

C: static void doStuff(int... doArgs, int y) { }

D: static void doStuff(int doArgs...) { }

8. What is coercion?

A: Coercion is a phenomenon of promoting sub class to super class

B: Coercion is a phenomenon of casting super class to sub class

C: Coercion is an changing the data types according to cast operator

**D: Coercion is the conversion between different data types done while compiling**

9. Which of the following statements is true?

- a) Java supports only procedural approach towards programming
- b) Both procedural and object oriented approach supported in java.
- c) Only object oriented approach supported in java.**
- d) Java does not support procedural approach.

10. Which command is used for execution of Java program?

- a. **Java**                      b. javac                      c. javap                      d. none of these

11. What would be the output of the following?

```
public class MyClass
{
 public static void main(String args[])
 {
 int i=257;
 byte b=(byte)i;
 System.out.println("b= "+b);
 }
}
```

- a) 1 b) -128 c) 257 d) compilation error

**Note:**

byte : Value range from -128 to 127.

u want to store "257" .

it will go up to 127 , [127]

now there is no 128.

It will start from -128 to 0 [129]

Total of 127+129 is 256

U still require 1

**After 0 u have 1, hence ans is 1.**

With this logic ,

int i=250;

byte b=(byte)i;

System.out.println("b= "+b);

Ans will be -6

12. After execution of the following code fragment, what are the values of the variable x, a and b?

int x, a = 6, b = 7;

x = a++ + b++;

1. x=15,a=7,b=8

2. x=15,a=6,b=7

**3. x=13,a=7,b=8**

4. x=13,a=6,b=7

13. Default value of reference type is

1. 0

2. /0

3. zero

**4. null**

14. Which declarations of main() method are valid in order to start the execution of an application?

a) public void main(String args[])

b) public void static main(String args[])

c) public static main(String args[])

**d) public static void main(String args[])**

15. int arr[]=new int[25];

Which statement is true?

**a) arr[24] is 0**

b) arr[24] is not defined

c) arr[25] is 0

d) arr[0] is null

16. You want to find out the value of the last element of an array. You write the following code. What will happen when you attempt to compile or run it?

```
Public class MyAr
{
Public static void main (String argv[])
{
int [] I = new int [5] ;
System.out.println(I [5]);
}
}
```

1. An error at compile time.
- 2. An error at run time.**
3. The value 0 will be output.
4. The string "null" will be the output

17. Wrapper class is part of package\_\_\_\_\_

- 1. java.lang**
2. java.util
3. java.io
4. java.awt

### GENERIC

- 1) At the time of compilation compiler removes all the information about generics. This is known as
  - A. Generic-removal
  - B. Generic-Erasure
  - C. Type-Erasure**
  - D. none of the above
- 2) <P extends Q> here Q can be either class or interface
  - A. True**
  - B. false
- 3) We can't have generic method in non-generic class
  - A. True
  - B. false**
- 4) Polymorphism applies to base type as well as generic type.
  - A. True
  - B. false**
- 5) Mixing generic and non-generics can be risky
  - A. True**
  - B. false
- 6) If the base class reference referring to sub class array then there is a possibility of
  - A. IllegalArgumentException
  - B. ArrayStoreException**
  - C. NullPointerException
  - D. none of the above
- 7) In case of <? Extends .....> we can add
  - A. True
  - B. false**

- 8) In case of `<? super .....` we can add  
**A. True** B. false
- 9) `List<? Super Thread> mylist=new ArrayList<Object>()` will work  
**A. Yes** B. no
- 10) `List<? Super Dog> mylist=new ArrayList<Animal>()` `mylist.add(new Cat());` will work  
A. Yes B. **no**
- 11) `List<?>` allows u to add  
A. True B. **false**
- 12) `List<Object>` allows u to add  
**A. True** B. false

### HIBERNATE

1. Hibernate entity can be  
A: transient  
B: detached  
C: persistent  
**D: all of the above**
2. When transaction completes, all the associated persistent objects still exists in memory but they lose their association with the session on encountering one of the following method  
A: `session.flush()`  
B: `trasaction.close()`  
**C: `session.close()`**  
D: none of the above
3. In hibernate, If the developer is not certain about the existence of the object.  
A: `load()` method should be used  
**B: `get()` method should be used**  
C: `retrieve()` method should be used  
D: all the above are same with not much difference.
4. One of the following is not an interface from Hibernate core interfaces.  
A: `SessionFactory`  
B: `Session`  
**C: `ConnectionFactory`**  
D: `Transaction`
5. Session interface does  
A: JDBC connection  
B: gets Transaction  
C: holds cache of persistent objects

**D: all of the above**

6. One of the following persist element index

A: <bag>

**B: <list>**

C: <set>

D: <map>

7. Which statement is wrong?

**A: HQL is case insensitive for class names and properties**

B: SQL operates on relations, HQL operates on objects.

C: HQL is ultimately generated into underlying SQL.

D: HQL supports polymorphic queries.

8. A join which combines result of both left and right outer join is

A: Left Right Join

B: Right Outer Join

**C: Full Join**

D: Left Join

9. Which class is used to create EntityManagerFactory ?

**A: Persistence**

B: EntityManager

C: EntityTranscation

D: EntityFactory

10. One of the following is ORM implementation

A: struts

B: JSF

**C: hibernate**

D: spring

11. After EntityManager is closed, entity is in a \_\_\_\_\_ state.

A: dead

**B: detached**

C: managed

D: persistent

12. What is the root tag element in the hibernate configuration file?

A: <hibernate-cfg>

B: <hibernate-conf>

**C: <hibemate-configuration>**

D: <hibemate>

13. Once the hibernate session is closed, in which state the object remains?

**A: detached**

B: transient

C: pesistent

D: garbage collector

14. What is the root level element in hibernate mapping file?

**a. <hibernate-mapping>**

b. <session-mapping>

c. <sessionfactory-mapping>

d. None of the above

15. What does <generator-class="native"/> means?

**a. Generate primary key**

b. Generate tables based on configuration

c. Generate sql statements on configuration

d. None

16. What does hibernate.hbm2ddl.auto.create this means?

**a. Create tables automatically**

b. Create session object automatically

c. Create session factory object automatically

d. None.

17. Which statement is correct about Hibernate?

**a. The CacheMode controls how a particular session interacts with the second-level cache.**

b. The CacheMode doesn't control session interaction with the second-level cache.

c. Both the above

d. None of the above

**18. When we integrate Hibernate via spring we don't need to take care of.**

A: SessionFactory

B: Session

**C: Both a and b. (since there is something called as "HibernateTemplate" in spring)**

D: None of the above.

19. What is dirty checking in Hibernate?

**a. object state changes in order to synchronize the updated state with the database**

b. Remove the dirty data from data base.

c. Check the data when insert into data base.

d. None

20. What is the root level element in a hibernate mapping file?

**a. <hibernate-mapping>**

b. <session-mapping>



- c. < sessionFactory-mapping >
  - d. none of the above
21. What does session.evict() method do?
- a. **remove the object and its collections from the first level cache**
  - b. remove the object and its collections from the second level cache
  - c. remove the object and its collections from the data base
  - d. None of the above
22. What does hibernate.hbm2ddl.auto create this means?
- a. create session object automatically
  - b. **create tables automatically**
  - c. create Session Factory object automatically
  - d. None
23. Is Hibernate Session threadsafe?
- a) Yes                      **b) no**                      c) no relation with thread                      d) none
24. Which 2nd level cache is better in hibernate?
- a. **EHCache**
  - b. GumoCache
  - c. DASCACHE
  - d. TestCache
25. Which statement is correct about session.load()?
- a. load() will return null if there is no matching database row.
  - b. **load() will throw an unrecoverable exception if there is no matching database row.**
  - c. None of the above.
26. Which of the following is tag of hibernate.cfg.xml?
- a. **SQL variant to generate**
  - b. Size of the database
  - c. All of the above
  - d. None of the above
27. How do we get a session object?
- a. sessionFactory.get();
  - b. (session)SessionFactory.getObject();
  - c. sessionFactory.getSession();
  - d. **SessionFactory.openSession();**
28. How to enable query cache in hibernate?
- a. hibernate.cache.query\_cache true
  - b. **hibernate.cache.use\_query\_cache true**
  - c. hibernate.cache.query\_cache yes
  - d. none

29. What does session.delete() do?
- remove from sessionFactory
  - remove data from in momery.
  - remove from database.**
  - All of the above.
30. The hibernate configuration file name is?
- http.conf
  - web.config
  - persistence.xml
  - hibernate.cfg.xml**
31. Load child object automatically when parent object is loaded?
- lazy=yes
  - lazy=false**
  - lazy=true
  - lazy=no
32. In order to create new session which function should be used ?
- A: getSession(true)  
B: getSession()  
**C: both A and B**  
D: getSession(false);
33. Which of the following is true?
- Hibernate is a framework
  - Hibernate is an Object-Relational mapping (ORM) library for the java language.
  - Hibernate provides mapping for object-oriented domain model to a traditional relational database.
  - All of the above.**
34. There are core interfaces that are used in just about every hibernate application. Using these interfaces, you can store and retrieve persistent object and control transactions. Select all the interfaces that you see.
- Configuration interface
  - Query and Criteria interfaces
  - All of the above**
  - None of the above
35. Which of the following is not hibernate session method?
- a) Save ()                      **b) remove ()**                      c) persist ()                      d) load ()
36. If the validation fails what will be returned by the validate () method?

a) Success

b) **input**

c) login

d) error

### INHERITANCE

1. What is the output of following code.

```
class a
{
 static
 {
 System.out.println(" static a");
 }
}
class b extends a
{
 static
 {
 System.out.println(" static b");
 }
}
class c extends b
{
 static
 {
 System.out.println(" static c");
 }
}
public class myclass
{
 static
 {
 System.out.println(" static
myclass");
 }
 public static void main(String args[])
 {
 new c();
 System.out.println("in main");
 }
}
```

A: in main, static a,static b,static c, static myclass

**B: static myclass, static a,static b,static c, in main**

C: static myclass, in main ,static a,static b,static c

D: static a,static b,static c, static myclass, in main ,

2. What will happen to the following code ?

```
class base
{
```

```
public final void disp ()
{
 System.out.println ("in disp");
}
}
public class sub extends base
{
 public static void main (String argv [])
 {
 base b = new base();
 b.disp () ;
 }
}
```

A: runtime error

B: compiler error "final method must be inside final class"

C: compiler error "a class having final method can not be inherited"

**D: neither compilation nor runtime error**

3. what will be the output ?

```
class base
{
 int i;
 base()
 {
 add(1);
 }
 void add(int v)
 {
 i+=v;
 }
 void print()
 {
 System.out.println(i);
 }
}
class sub extends base
{
 sub()
 {
 add(2);
 }
 void add(int v)
 {
```

```
 i+=v*2;
 }
}
public class test6
{
 static void disp(base b)
 {
 b.add(8);
 b.print();
 }
 public static void main(String args[])
 {
 disp(new sub());
 }
}
```

- A: 9
- B: 18
- C: 22**
- D: 21

4. What is the output of following code ?

```
interface emp
{
}
public class Trial implements emp
{
 public static void main(String args[])
 {
 Trial t=new Trial();
 if(t instanceof Trial)
 {
 System.out.println("Trial");
 }
 if(t instanceof emp)
 {
 System.out.println("emp");
 }
 if(t instanceof Object)
 {
 System.out.println("Object");
 }
 }
}
```

- A: Trial, emp, Object**
- B: Trial, emp
- C: compilation error "can not use instanceof with interface"

D: Trial, Object

5. what is the output of the following code?

```
class a
{
 static
 {
 System.out.println("static a");
 }
}
class b extends a
{
 static
 {
 System.out.println("static b");
 }
}
class c extends b
{
 static
 {
 System.out.println("static c");
 }
}
public class MyClass
{
 static
 {
 System.out.println("static MyClass");
 }
 public static void main(String args[])
 {
 new c();
 System.out.println("in main");
 }
}
```

- a) in main, static a, static b, static c, static MyClass
- b) static MyClass, static a, static b, static c, in main**
- c) static MyClass, in main, static a, static b, static c
- d) static a, static b, static c, static MyClass, in main

6. what will happen to the following code?

```
class base
{
 public final void disp()
 {
 System.out.println("disp");
 }
}
```

```
 }
}

public class sub extends base
{
 public static void main(String args[])
 {
 base b=new base();
 b.disp();
 }
}
```

- a) runtime error
- b) compiler error: final method must there in final class
- c) compiler error: a class having final method can not be instantiated.
- d) Neither compile time nor runtime error.**

7. Why multiple inheritance is not available in java?

- a. It leads to confusion for a Java program**
- b. The programmer can achieve multiple inheritance by using interface
- c. The programmer can achieve multiple inheritance by repeatedly using single inheritance
- d. All of the above

8. what is the output?

```
class base
{
}
class sub1 extends base
{
}
class sub2 extends sub1
{
}
class sub3 extends sub2
{
}
public class test12
{
 public static void main(String args[])
 {
 sub1 s=new sub2();
 base b=s;
 if(b instanceof base)
 {
 System.out.println("base");
 }
 if(b instanceof sub1)
 {
 System.out.println("sub1");
 }
 }
}
```

```

 }
 if(b instanceof sub2)
 {
 System.out.println("sub2");
 }
 if(b instanceof sub3)
 {
 System.out.println("sub3");
 }
}

```

```

}

```

- a. base                      b. **sub3**                      c. sub1                      d. sub2

9. Given the following code, what can be said about the statement `s=(sub)b` ?

```

class base
{
}
class sub extends base
{
}
public class test12
{
 public static void main(String args[])
 {
 base b=new base();
 sub s=new sub();
 s=(sub)b;
 }
}

```

- a. **legal at compile time but illegal at runtime**  
 b. illegal at compile time  
 c. legal at compile and runtime ,but (sub) cast is not needed  
 d. legal at compile and runtime ,but (sub) cast is strictly needed.

10. What will happen when you attempt to compile or run this code?

```

class Base
{
 public final void amethod ()
 {
 system.out.println ("amethod");
 }
}
public class Fin extends Base
{
 public static void main (String argv [])
 {

```



```
Base b = new Base();
b.amethod();
}
}
```

- a. Compile time error indicating that a class with any final methods must be declared final itself
- b. Compile time error indicating that you inherit from a class with final methods.
- c. Run time error indicating that Base is not defined as final.
- d. Success in compilation and output of “amethod” at run time**

**11.** class Foo

```
{
 int num;
 Bar comp=new Bar();
}
class Bar
{
 boolean flag;
}
class Baz extends Foo
{
 Bar thing=new Bar();
 double d;
}
```

- a. A Bar is a Baz
- b. A Foo has a Bar**
- c. A Baz is a Foo**
- d. A Foo is a Baz
- e. A Baz has a Bar.**

**12.** What will happen to the following code?

```
interface X
{
 static void disp()
 {
 System.out.println("in disp of X");
 }
}
```

```
public class Trial implements X
{
 public static void main(String args[])
 {
 Trial t=new Trial();
 t.disp();
 }
}
```

```
}
}
```

- a. Compilation error “disp not available with Trial”
- b. Compilation error “static method can not be defined inside an interface”
- c. Compilation error “ Trial class must define disp as it is there inside parent interface”
- d. Output “ in disp of X”

13. Given

interface emp // functional interface

```
{
 String wish(String name);
}
```

Lambda expression in order to use above interface would be:

- a. emp ref2=(String name)->{ return "Welcome to our site\t"+name;;}
- b. emp ref2=(String name){ return "Welcome to our site\t"+name;;}**
- c. Both A and B
- d. None of the above

14. How restrictive is the default accessibility compared to public, protected and private accessibility?

- a. Less restrictive than public.
- b. More restrictive than public, but less restrictive than protected
- c. More restrictive than private
- d. More restrictive than protected, but less restrictive than private**
- e. Less restrictive than protected from within a package, and more restrictive than protected from outside a package

15. What will be the output of the following code?

```
public class VerySmart
{
 public static void main(String[] args)
 {
 String message;
 System.out.println("message length is : " +
 message.length());
 }
}
```

- a. /0
- b. 0
- c. compile time error**
- d. run time error

16. The programmer must explicitly create the System.in and System.out objects.

**A. True**

**B. False**

17. A method within a class is only accessible by classes that are defined within the same package as the class of the method. How can such a restriction be enforced?

- A. Declare the method with the keyword "public"
- B. Declare the method with the keyword "protected"
- C. Do not declare the method with any modifiers.**
- D. Declare the method with the keyword "private"
- E. Declare the method with the keyword "package"

18. A final class cannot have any abstract methods.

**A. True**

**B. False**

19. String class is

- a. final**
- b. abstract
- c. static
- d. transient

20. what is the result of following code ?

```
class base
{
 int i;
 base()
 {
 add(1);
 }
 void add(int v)
 {
 i+=v;
 }
 void print()
 {
 System.out.println(i);
 }
}
class sub extends base
{
 sub()
 {
 System.out.println("in sub def const");
 super.add(2);
 }
 void add(int v)
 {
 i+=v*2;
 }
}
public class test11
{
 public static void main(String args[])
 {
 base b;
```

```
 b=new sub();
 b.print();
 }
}
```

a. 4  
b. 3  
c. Error: super has to be on first line of constructor  
d. 2

21. What is garbage collection process in java?

- a. The operating system periodically deletes all the java files available on the system.
- b. Unused package in program is automatically deleted.
- c. **When all references to an object are gone, memory used by that object is automatically reclaimed.**
- d. The JVM checks the output of any java program and deletes anything that does not make sense.

22. Given the following code,

```
public class Test
{
 String str="hello";
}
```

- 1. Test t=new Test();
- 2. System.out.println(t.str);
- 3. t=null;
- 4. System.out.println(t.str);
- 5. System.out.println("done");

What will happen to the above code?

- A: "NullPointerException" at Line 3
- B: "NullPointerException" at Line 4**
- C: Compilation error at Line 4
- D: Successful out

23. Given the following code,

```
public class Test
{
 String str="hello";
}
```

- 1. Test t=new Test();
- 2. System.out.println(t.str);
- 3. t.str=null;
- 4. t=null;
- 5. System.out.println("done");

At which line the object created at 1 will be marked for garbage collection?

- A: Line 3
- B: Line 4**
- C: Can't say exactly when

D: both Line3 and Line4

24. What is the output?

```
public class Trial
{
 int num=10;
 void change(Trial ref)
 {
 ref.num=20;
 ref=new Trial();
 ref.num=30;
 ref=null;
 }
 public static void main(String args[])
 {
 Trial t=new Trial();
 t.change(t);
 System.out.println(t.num);
 }
}
```

A: 30

**B: 20**

C: NullPointerException

D: 10

25. class Bar { }

class Test

```
{
 Bar doBar()
 {
 Bar b = new Bar(); /* Line 6 */
 return b; /* Line 7 */
 }
 public static void main (String args[])
 {
 Test t = new Test(); /* Line 11 */
 Bar newBar = t.doBar(); /* Line 12 */
 System.out.println("newBar");
 newBar = new Bar(); /* Line 14 */
 System.out.println("finishing"); /* Line 15 */
 }
}
```

At what point is the Bar object, created on line 6, eligible for garbage collection?

A. after line 12

**B. after line 14**

C. after line 7, when doBar() completes

D. after line 15, when main() completes

26. What is the output for the following program?

```
class A
{
 static
 {
 System.out.println("in A static block");
 }
}
public class Trial
{
 A ob=new A();
 public static void main(String args[])
 {
 System.out.println("in main");
 }
 static
 {
 System.out.println("in Trial static block");
 }
}
```

A: in A's static block, in Trial static block, in main

**B: in Trial static block, in main**

C: in A's static block, ,in main ,in Trial static block

D: in Trial static block, in A's static block, in main

27. Given following code, what will happen to it ?

```
String str1="hello";
String str2="hel";
String str3=str2+"lo";
if(str1==str3)
{
 System.out.println("str1 and str3 are==");
}
else
{
 System.out.println("str1 and str3 are not ==");
}

if(str1.equals(str3))
{
 System.out.println("str1 and str3 are equals");
}
else
{
 System.out.println("str1 and str3 are not equals");
}
```

```
System.out.println("str1 and str3 are not equals");
```

```
}
```

A: str1 and str3 are ==, str1 and str3 are equals

**B: str1 and str3 are not ==, str1 and str3 are equals**

C: str1 and str3 are ==, str1 and str3 are not equals

D: compilation error

28. Java supports

A. single level inheritance

B. multi-level inheritance

C. hierarchical inheritance

**D. all of the above**

29. Super must be on first line if we want to invoke base class constructor.

**A. True**

B. False

30. Super need not be on first line if we want to invoke base class method.

**A. True**

B. False

31. <default> is more accessible than protected.

A. True

**B. False**

32. Final keyword can be applied to

A. Instance member

B. Class variable

C. Local variable

**D. All of the above**

33. In java we can apply static modifier for local variable.

A. True

**B. False**

34. In order to make a class abstract:

**A. Apply abstract keyword to class**

B. Declare abstract method inside class

C. Both a and b

D. None of the above

35. In order to check "is-a" relationship, we use following operator

A. Is-a

**B. instanceof**

C. Is\_relationship

D. None of the above

36. If we try to cast the classes out of hierarchy we get

A. BadCastException

B. OutOfHierarchyException

**C. ClassCastException**

D. None of the above

37. At the time of overriding function, if we change the argument :

A. It gives compiler error

- B. It gives runtime error
- C. Compiler automatically removes the argument
- D. It becomes overloading.**

38. Will following code work?

Class MyClass extends String{}

- A. Yes
- B. No**

### JAVA FX

1. In JavaFX following class is acting as a container for all the contents

- a. Scene**
- b. Stage
- c. LayoutPane
- d. None of the above

2. In order to start every JavaFX application you must invoke following method

- a. Init()
- b. Start()
- c. Launch()**
- d. None of the above

### JDBC

1. Which of the following code will you use to get a count of the columns in the result?

- A.     ResultMetaData rsmd=DatabaseMetaData.getMetaData();  
int columns=rsmd.getColumnCount();
- B.     ResultSetMetaData rsmd=new ResultSetMetaData(result);  
int columns=rsmd.getColumnCount();
- C.     ResultSetMetaData rsmd=result.getMetaData();  
int columns=rsmd.getColumnCount();**
- D.     DatabaseMetaData md=result.getMetaData();  
int columns=md.getColumnCount();

2. The Jdbc driver that directly communicates with database protocol is

- a. Type I
- b. Type II
- c. Type III
- d. Type IV**

3. What will the following line of code do? Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

- a. Load the Driver for Database access**
- b. Establish a connection with the specified database
- c. Accesses data from a table
- d. Create a ResultSet object

4. Which character is used to represent an input parameter in a PreparedStatement?

- a. %
- b. \*
- c. ?**



d. #

5. Which of the following statements is NOT true about a PreparedStatement

- a. PreparedStatement is pre-compiled
- b. PreparedStatement may have both IN and OUT parameters.**
- c. Execution of PreparedStatement is faster than Statement.
- d. All of the above

6. Which class gives u connection from connection pool?

A: Driver

**B: DataSource**

C: DriverManager

D: Connection

7. Which driver can be used for internet purpose?

A: Type 1

B: Type 2

**C: Type 3**

D: all the above

8. Which class gives connection?

A: Driver

B: Statement

**C: DriverManager**

D: Connection

9. According to JDBC4, driver class name should be there inside

A: META-INF\java.sql.Driver file

B: Services\java.sql.Driver file

C: Services\META-INF\java.sql.Driver file

**D: META-INF\services\java.sql.Driver file**

10. If we use query statement inside "execute()" method of Statement, it will return

A: false

**B: true**

C: 1

D: 0

11. If we use update statement inside "execute()" method of Statement, it will return

**A: false**

B: true

C: 1

D: 0

12. When driver is not found, it results in

A: DriverNotFoundException

**B: ClassNotFoundException**

C: SQLException

D: None of the above.

13. Connection is

**A: interface**

B: class

C: package

D: None of the above.

14. Name the most suitable execution method in JDBC , for firing DML queries.

a. executeQuery()

**b. executeUpdate()**

c. executeQueue()

d. executeDynamicQuery()

15. in the following code sample

```
Class.forName("Oracle.jdbc.OracleDriver");
```

```
String dbUrl="jdbc:oracle:thin:@host1:mydb";
```

```
Connection con=DriverManager.getConnection(dbUrl,"scott","tiger");
```

How to create PreparedStatement in order to support Scrollable and Read-only ResultSet?

a. con.prepareStatement(sql,TYPE\_SCROLL\_INSENSITIVE,CONCUR\_UPDATABLE);

b. con.prepareStatement(sql,TYPE\_FORWARD\_ONLY,CONCUR\_UPDATABLE);

**c. con.prepareStatement(sql,TYPE\_SCROLL\_INSENSITIVE,CONCUR\_READ\_ONLY);**

d. con.prepareStatement(sql,TYPE\_SCROLLABLE,CONCUR\_UPDATABLE);

16. PreparedStatements are actually \_\_\_\_\_

**a) Compiled**

b) not compiled

c) They don't compile but directly get executed

d) none of the above

17. Which driver is efficient and always preferable for jdbc applications?

a) **Type – 4**

b) Type –1

c) Type –3

d) Type –2

18. The parameters of PreparedStatement object are \_\_\_\_\_ when the user clicks on the query button.

a) **Initialized**

b) started

c) paused

d) stopped

19. Name of Type4 driver is

**A. Native-Protocol Driver**

B. JDBC-ODBC bridge

C. Network-Protocol Driver

D. Native-API Driver

20. By default ResultSet is

- A. None of these
- B. TYPE\_SCROLL\_INSENSITIVE
- C. TYPE\_FORWARD\_ONLY**
- D. TYPE\_SCROLL\_SENSITIVE

21. PreCompiled SQL statements are stored in

- A. None of these
- B. CallableStatement
- C. Statement
- D. PreparedStatement**

22. The checked exception which we need to handle in case of jdbc is

- A. None of these
- B. SQLException**
- C. IOException
- D. JDBCException

23. Type1 driver makes use of

- A. None of these
- B. Odbc**
- C. Jdbc
- D. Middleware

24. If ResultSet is Scrollable we can invoke

- A. Absolute
- B. Next
- C. Previous
- D. all of the above**

25. Stored Procedures can be called using

- A. PreparedStatement
- B. Statement
- C. CallableStatement**
- D. None of these

26. One of the following is a class

- a) Connection
- b) Statement
- c) ResultSet
- d) none of the above**

27. Every driver class has \_\_\_\_\_ which registers itself with the DriverManager

- a) Static method
- b) static block**
- c) constructor
- d) none of the above

### JSF

1. Front Controller in JSF is
  - a) DispatcherServlet
  - b) ActionServlet
  - c) FacesServlet**
  - d) none of the above
  
2. Configuration file in JSF is
  - a) webconfig.xml
  - b) jsf-config.xml
  - c) applicationContext.xml
  - d) none of the above [it is "faces-config.xml"]**
  
3. Annotation which is used to define bean in JSF is
  - a) @managedBean**
  - b) @Bean
  - c) @Component
  - d) @JSFBean
  
4. In case of JSF component tree is stored inside
  - A. ServletContext
  - B. ApplicationContext
  - C. FacesContext**
  - D. None of the above

### JSP

1. Which of the attribute of JSP mentioned below is not available in servlet?
  - A: request
  - B: session
  - C: page**
  - D: context
  
2. A method can be defined inside
  - A: JSP scriptlet
  - B: JSP declaration**
  - C: JSP expression
  - D: none of the above
  
3. One of the following cannot be overridden while writing JSP page
  - A: \_jspService**
  - B: jsplnit

C: JspDestroy

D: JspService

4. Which attribute cannot be used inside <jsp:useBean>

A: class

B: id

**C: name**

D: scope

5. If we want to include some dynamic page in JSP we should use

**A: <jsp:include page="/somepage"/>**

B: <%@ include file="/somefile" %>

C: RequestDispatcher rd=request.getRequestDispatcher("/somepage");  
rd.include(request,response);

D: none of the above

6. One of the following JSTL tag performs URL rewriting.

A: link

**B: url**

C: aHref

D: import

7. Given

```
<jsp:useBean id="a1" scope="request" class="mypack.Customer"/>
```

What is the syntax to read Customer object?

A: \${requestScope.a1}

B: <%=request.getAttribute("a1")%>

C: \${a1}

**D: all the above**

8. Which JSP expression tag will print the context initialization parameter named "uname"?

**A: <%= application.getAttribute("uname")%>**

B: <%= application.getInitParameter("uname")%>

C: <%= request.getParameter("uname")%>

D: <%=contextParam.get("uname")%>

9. When a JSP page is compiled, what is it turned into?

a. Applet

**b. Servlet**

c. Application

d. Maillet

10. The implicit JSP objects like request, response, and out are only visible in the \_jspService() method.

**a. True**

**b. False**

11. To fill up all bean properties with HTML form elements which of the following statements can be used,
- a. `<jsp:synchronize name="BeanName" />`
  - b. `<jsp:setProperty name="BeanName" property="" />`
  - c. **`<jsp:setProperty name="BeanName" property="*" />`**
  - d. `<jsp:setProperty name="BeanName" property="All" />`
12. Why use `RequestDispatcher` to forward a request to another resource, instead of using a `sendRedirect`?
- a. Redirects are no longer supported in the current servlet API.
  - b. Redirects are not a cross-platform portable mechanism
  - c. The `RequestDispatcher` does not use the reflection API.
  - d. **The `RequestDispatcher` does not require a round trip to the client, and thus is more efficient and allows the server to maintain request state.**
13. The attribute which defines your jsp page as a exception handling page is
- a. `isExceptionPage`
  - b. `exceptionPage`
  - c. **`isErrorPage`**
  - d. `w. errorPage`
14. To use the `ServletContext` in Jsp the implicit object is
- a. **application**
  - b. context
  - c. config
  - d. page
15. Which attribute can be used inside `<jsp:setProperty>` and `<jsp:getProperty>`
- A: class  
B: id  
**C: name**  
D: scope
16. What is the role of J2EE container in web application?
- A: It provide the life-cycle.  
B: It compiles the JSP.  
C: Loads the classes of the JSP and servlet.  
**D: All of the above**
17. How can we enable session tracking for JSP PAGES?
- A: Enable cookies in the browser  
**B: set session attribute value is true in page directive**  
C: set is Session attribute true in page directive

D: Enable session tracking in the browser

18. What is the effect of executing the following JSP statement, assuming a class with name Employee exists in class's package?

```
<%@ page import = "classes.Employee" %>
```

```
<jsp:useBean id="employee" class="classes.Employee" scope="session"/>
```

```
<jsp:setProperty name="employee" property="*" />
```

A: The code does not compile as there is no properly attribute of setProperty tag.

B: The code does not compile as property attribute cannot take \* as a value.

C: The code sets value of all properties of employee bean to "\*".

**D: The code sets the values of all properties of employee bean to matching parameters in request object.**

19. If a JSP page overrides the jsplnit() method using a declaration JSP tag, which phase of the JSP page life-cycle generates the overridden method in the servlet?

**A: page translation.**

B: JSP page compilation.

C: call jsplnit()

D: call\_jspService().

20. <jsp:useBean id="a1" scope="request" class="mypack.Customer" ./>

What is the syntax to read Customer object?

A: \${requestScope.a1}

B: <%=request.getAttribute("a1")%>

C: \${a1}

**D: All the above**

21. Which method can be overridden while writing JSP page

a. jspInit

b. JspDestroy

**c. A and B both**

d. None of the above

22. The implicit object "out" of Jsp is of type

a. PrintWriter

b. InputStream

**c. JspWriter**

d. JspPrintWriter

23. Two of the following are the attributes of <%taglib%> directive.

a) Name

**b) prefix**

c) value

**d) uri**

e) url

24. The attribute which informs container about a particular error jsp page is

- A. isExceptionPage
- B. exceptionPage
- C. isErrorPage
- D. errorPage**

25. Which of the following is true for JavaBean?

- a) It can not be a GUI component.
- b) It never implement Serializable interface
- c) It has Zero argument constructor**
- d) It is a distributed component

26. if a jsp page overrides the jsplnit method using a declaration JSP tag , which phase of the jsp page life-cycle generates the overridden method in servlet?

- a) Page translation**                      b) JSP page compilation                      c) call jsplnit()                      d) call \_jspService()

27. Select the correct statement about following code (Select one)

```
<%@page language="java"%>
```

```
<html><body>
```

```
<%
```

```
response.getOutputStream().print("hello");
```

```
out.print("World");
```

```
%>
```

```
</body></html>
```

- a. It will print “hello World” in the output
- b. It will generate compile time errors
- c. It will throw runtime exceptions (java.lang.IllegalStateException: getOutputStream() has already been called for this response)**
- d. It will only print “hello”
- e.

28. Which is not a valid scope for javabean in JSP?

- a) Page                      b) session                      c) request                      d) **global**

29. Which among the following will compile?

- a) <% int x=10 %>
- b) <%= “hello how are you” %>**
- c) <%= “hello” ; %>
- d) <%! Int x=10 %>

30. Which of the following statements is true regarding the scope of “request” in JSP?

- a. Objects with request scope are accessible from pages processing the same request where they were created.



- b. All references to the object shall be released after the request is processed: in particular if the request is forwarded to a resource in the same runtime , the object is still reachable.
- c. References to the objects in request scope are stored in a request object.
- d. **All of the above.**

31. Which of the following statements makes your compiled JSP page implement the SingleThreadModel interface?

- a. `<%@ page isThreadSafe="false" %>`
- b. `<%@ page isThreadSafe="true" %>`

32. How can a servlet call a JSP error page?

- a. This capability is not supported
- b. When the servlet throws the exception, it will automatically be caught by the calling JSP page.
- c. **The servlet needs to forward the request to the specific error page URL. The exception is passed along as an attribute named "javax.servlet.jsp.jspException"**
- d. The servlet needs to redirect the response to the specific error page, saving the exception off in a cookie.

33. When using a JavaBean to get all the parameters from a form, what must the property be set to (??? in the following code) for automatic initialization?

```
<jsp:useBean id="fBean" class="govt.FormBean" scope="request"/>
<jsp:setProperty name="fBean" property="???" />
<jsp:forward page="/servlet/JSP2Servlet" />
```

- a. \*
- b. All
- c. @
- d. =

34. Choose the statement that best describes the relationship between JSP and servlets:

- a. Servlets are built on JSP semantics and all servlets are compiled to JSP pages for runtime usage.
- b. JSP and servlets are unrelated technologies.
- c. Servlets and JSP are competing technologies for handling web requests. Servlets are being superseded by JSP, which is preferred. The two technologies are not useful in combination.
- d. **JSPs are built on servlet semantics and all JSPs are compiled to servlets for runtime usage.**

35. What alternatives exist to embedding Java code directly within the HTML markup of your JSP page?

- a. Moving the code into your session manager.
- b. **Moving the code into scriptlets.**
- c. Moving the code into JavaBeans and servlets
- d. Moving the code into a transaction manager

36. What is the initial contact point for handling a web request in a *Page-Centric* architecture?

- a. **A JSP page**
- b. A JavaBean.
- c. A servlet.
- d. A session manager

37. Which object would you use to share user specific information between JSPs?

- a. Request                      b. Response                      c. **Session**                      d. Application

38. Which of the following rules must a reusable JavaBeans™ component adhere to?

- a. **The Bean class must provide zero argument constructors.**
- b. The Bean must have a corresponding BeanInfo class.
- c. The Bean must only use visible components.
- d. The Bean must not be serializable.

39. Which of the following is not a standard method called as part of the JSP life cycle?

- a. jspInit()
- b. **jspService()**
- c. \_jspService()
- d. jspDestroy()

40. If you want to override a JSP file's initialization method, within what type of tags must you declare the method?

- a. <@ @>
- b. <%@ %>
- c. <% %>
- d. **<%! %>**

41. Which of the following cannot be used as the scope when using a JavaBean with JSP?

- a. Application
- b. Session
- c. Request
- d. **Response**
- e. Page

42. What is the key difference between using a <jsp:forward> and HttpServletResponse.sendRedirect()?

- a. forward executes on the client while sendRedirect() executes on the server
- b. **Forward executes on the server while sendRedirect() executes on the client.**
- c. The two methods perform identically.

43. Two of the following methods can be overridden in jsp page

- a) **jspInit**                      b) \_jspService                      c) jspGet                      d) doPost                      e) **jspDestroy**

44. One of the following is jsp expression

- a) `<%! %>`      b) `<% %>`      c) `<%@ %>`      d) `<%= %>`

45. One of the following is not an implicit jsp object

- a) **context**      b) config      c) session      d) pageContext

46. One of the following is an additional attribute in Jsp as compare to servlet attributes .

- a) Request      b) context      c) **page**      d) response      e) session

47. Two of the following are used to call bean class setter and getter methods.

- A) setAttribute      b) **getProperty**      c) getAttribute      d) **setProperty**

48. One of the following is not a jsp standard action

- a) forward      b) include      c) **page**      d) param

49. When jsp is generated into servlet it is derived from which class

- a) HttpServlet      b) **HttpJspBase**      c) HttpJspPage      d) Servlet

50. Pure java code can be written inside

**A: JSP scriptlet**

B: JSP declaration

C: JSP expression

D: none of the above

51. For every tag encounter, a new instance of Tag Class is created.

- a. **True**      b. False

52. One of the following JSTL tag performs URL rewriting Select one:

- A. **url**  
B. aHref  
C. import  
D. link

53. Generated servlet in case of JSP extends \_\_\_\_.

- A. None of these  
B. **HttpJspBase**  
C. Service  
D. HttpServlet

54. Which attribute cannot be used inside `<jsp:useBean>`

- a. scope  
b. **name**  
c. id  
d. class

55. One of the following cannot be overridden while writing JSP page

- A. `jspService`
- B. `JspDestroy`
- C. `jspInit`
- D. `_jspService`

56. All EL implicit objects are internally

- A. **Map**
- B. List
- C. Set
- D. none of these

### MULTITHREADING

1. One of the following method is not executed by the programmer while writing multithreaded applications.

- A: start
- B: sleep
- C: join
- D: run**

2. Given

```
public class Trial extends Thread
{
 public void run()throws NullPointerException
 {
 System.out.println("hello");
 }
 public static void main(String args[])
 {
 new Trial().start();
 System.out.println("done");
 }
}
```

- A: `NullPointerException` during runtime
- B: Compilation error "overridden method does not throw `NullPointerException`"
- C: output "done" "hello"**

D: it will print "done" and then throw "NullPointerException"

3. Which of the following is the wrong statement

A: you cannot notify a particular thread

B: synchronized keyword can be applied to static methods

C: wait, notify methods can be called only from synchronized methods or block

**D: InterruptedException is unchecked exception.**

4. The \_\_\_\_\_ interface should be implemented by any class whose instances are intended to be executed by a thread.

A: Serializable

B: Comparable

C: Collection

**D: Runnable**

5. Consider the following:

class X implements Runnable

```
{
public static void main(String args[])
{
/* Missing code? */
}
public void run() { }
}
```

Which of the following lines of code is suitable to start a thread?

A: Thread t= new Thread(X);

B: Thread t= new Thread(X); t.start();

**C: X run = new X(); Thread t= new Thread(run); t.start();**

D: Thread t= new Thread(); x.run();

6. Which of the following statements is true?

A: A static method cannot be synchronized

B: Non-synchronized method can become synchronized if it's being called from a synchronized method

**C: When a thread call wait() from a synchronized method, it releases the lock**

D: Primitive variables can be protected from concurrent access using synchronized block.

7. Given

```
1. public class TestOne {
2. public static void main (String[] args) {
3. Thread.sleep(3000);
4. System.out.println("sleep");
```

5. }

6. }

A: No error, prints sleep

**B: Compilation error**

C: Runtime Error

D: No error & no output

8. Which of the following are methods of the Runnable interface?

**A: run**

B: start

C: yield

D: stop

9. While using Thread, which is incorrect

a. **u invoke run()** - correct ans

b. u invoke start()

c. u implement Runnable

d. u extend Thread

10. Which type of instanceof does targetObject have to pass for this to be legal while using

Thread t=new Thread(targetObject);

a) targetObject instanceof Thread

b) targetObject instanceof Applet

c) targetObject instanceof Object

**d) targetObject instanceof Runnable**

11. \_\_\_\_\_ are utilized to control the access to an object especially in multithreaded programming?

a) Asynchronized methods    b) serialized methods    c) **synchronized methods**    d) both a and c

12. \_\_\_\_\_ means each method in multithreaded environment doesn't access data by multiple threads at the same time.

a) Thread detach

b) thread isolation

c) **thread safety**

d) thread lock

13. Which of the following starts the default thread available in java program?

a) System class    b) **main method**

c) static keyword

d) none of these

14. Which two can be used to create a new Thread?

**A. Extend java.lang.Thread and override the run method.**

B. Extend java.lang.Runnable and override the start method.

C. Implement java.lang.thread and implement the run method.

**D. Implement java.lang.Runnable and implement the run method.**

15. What is the use of the synchronized keyword?

a. Allows two process to run in parallel but to communicate with each other

**b. Ensures only one thread at a time may access a method or object**

- c. Ensures that two or more processes will start and end at the same time
- d. Ensures that two or more Threads will start and end at the same time

16. What will happen when you attempt to compile and run the following code?

```
public class Bground extends Thread
{
 public static void main(String argv[])
 {
 Bground b = new Bground(); b.run();
 }
 public void start()
 {
 for (int i = 0; i<10; i++)
 {
 System.out.println("Value of i = " + i);
 }
 }
}
```

- a. A compile time error indicating that no run method is defined for the Thread class
- b. A run time error indicating that no run method is defined for the Thread class
- c. Clean compile and at run time the values 0 to 9 are printed out
- d. Clean compile but no output at runtime**

17. Given the following,

```
1. class MyThread extends Thread {
2.
3. public static void main(String [] args) {
4. MyThread t = new MyThread();
5. t.start();
6. System.out.print("one. ");
7. t.start();
8. System.out.print("two. ");
9. }
10.
11. public void run() {
12. System.out.print("Thread ");
13. }
14. }
```

What is the result of this code?

- A. Compilation fails
- B. An exception occurs at runtime. java.lang.IllegalThreadStateException**
- C. Thread one. Thread two.
- D. The output cannot be determined

18. What is the o/p of the following program?

```
1. class MyThread extends Thread {
2.
3. public static void main(String [] args) {
4. MyThread t = new MyThread();
5. Thread x = new Thread(t);
6. x.start();
7. }
8.
9. public void run() {
10. for(int i=0;i<3;++i) {
11. System.out.print(i + "..");
12. }
13. }
14. }
```

Compilation fails.

- a. 1..2..3..
- b. 0..1..2..3..
- c. **0..1..2..**

19. In case of class lock, non-static synchronized methods come into picture.

- a) **False**
- b) true

20. Sleep releases the lock whereas wait does not.

- a. True
- b. **False**

21. What is the effect of issuing a wait () method on an object

- a) 1. If a notify() method has already been sent to that object then it has no effect
- b) **2. The object issuing the call to wait() will halt until another object sends a notify() or notifyAll() method**
- c) 3. An exception will be raised
- d) 4. The object issuing the call to wait() will be automatically synchronized with any other objects using the receiving object.

22. One of the following method has to be invoked by the programmer in order to bring thread from born to runnable state.

- A: start**
- B: sleep
- C: join
- D: run

23. Which of the following is the correct statement



A: you can not notify a particular thread

B: synchronized keyword can be applied to static methods

C: wait, notify methods can be called only from synchronized methods or block

**D: all of the above.**

24. Select the correct statement:

**A. in case of intrinsic lock, when exception is raised in a synchronized code, lock is automatically released.**

B. in case of Reentrant lock, when exception is raised lock is automatically released.

C. Both A and B.

D. None of these.

25. Threads are lightweight as compare to processes

**A. True**

B. false

26. The method used to register thread with JVM scheduler

A. run

B. register

C. **start**

D. none of the above

27. By default the priority of thread is

A. Minimum

B. maximum

C. **normal**

D. none of the above

28. Sleep releases the lock wait does not

A. True

**B. false**

29. One of the following methods programmer never invokes in case of multi-threading application

A. **Run**

B. start

C. wait

D. notify

30. We can invoke wait, notify or notify all from non-synchronized methods

A. True

**B. false**

31. What will happen?

```
public class MyThread extends Thread
{
 @Override
 public void start()
 {
 }
}
```

```
public static void main(String args[])
{
 MyThread m1=new MyThread();
 m1.run();
}
}
```

- A. Compile time error
- B. Exception during runtime
- C. No error no output**
- D. Program will behave differently on different platforms

32. Wait, notify and notifyAll methods are

- A. Abstract
- B. static
- C. final**
- D. none of the above

33. All the blocking methods i.e. sleep, wait and join can throw

- A. IllegalMonitorStateException
- B. InterruptedException**
- C. BlockingException
- D. none of the above

34. What will happen?

```
class MyTarget implements Runnable
{
 public void run()
 {
 System.out.println("MyTarget run");
 }
}
public class MyApp
{
 public static void main(String args[])
 {
 MyTarget m=new MyTarget();
 Thread t1=new Thread();
 t1.start();
 }
}
```

- A. Output "MyTarget run"
- B. No output**
- C. Compilation error
- D. IllegalMonitorException during runtime

35. What will happen?

```
class MyTarget implements Runnable
{
 public void run()
 {
 System.out.println("MyTarget run");
 }
}
public class MyApp
{
 public static void main(String args[])
 {
 MyTarget m=new MyTarget();
 Thread t1=new Thread();
 t1.start(m);
 }
}
```

- A. Output "MyTarget run"
- B. No output
- C. Compilation error**
- D. IllegalMonitorException during runtime

36. What will happen?

```
class MyTarget implements Runnable
{
 public void run()
 {
 System.out.println("MyTarget run");
 }
}
public class MyApp
{
 public static void main(String args[])
 {
 MyTarget m=new MyTarget();
 Thread t1=new Thread(m);
 t1.start();
 }
}
```

- A. Output "MyTarget run"**
- B. No output
- C. Compilation error
- D. IllegalMonitorException during runtime

37. A class which contains non-static synchronized methods or blocks is called as \_\_\_\_\_

- A. Singleton
- B. Synchronized

C. **Thread-Safe**

D. none of the above

38. \_\_\_\_\_ method makes caller thread wait till this thread die.

- a. Wait
- b. sleep
- c. yield
- d. **join**

### MVC

1. One of the following is responsible for responding to user input and perform interactions on the data model objects.

A: model

B: view

**C: controller**

D: none of them

2. In MVC architecture model is \_\_\_\_\_

a) jsp

**b) javabean**

c) servlet

### Oops

1. What is the output?

```
public class Trial
{
 int num=10;
 void change(Trial ref)
 {
 ref.num=20;
 ref=null;
 }
 public static void main(String args[])
 {
 Trial t=new Trial();
 t.change(t);
 System.out.println(t.num);
 }
}
```

**A: 20**

B: 10

C: NullPointerException

D: None of the above

2. Which of the following modifiers can be applied to Top Level classes?

A: public

B: default

C: protected

**D: both A and B**

3. Which is true about an anonymous inner class?

A. It can extend exactly one class and implement exactly one interface.

B. It can extend exactly one class and can implement multiple interfaces.

**C. It can extend exactly one class or implement exactly one interface.**

D. It can implement multiple interfaces regardless of whether it also extends a class.

4. Local inner class cannot access

A: outer class member

B: its own static member

**C: local members of the method in which it is defined**

D: static member of outer class

5. Given

```
public static void main(String args[])
{
 Integer i;
 if(i==65)
 {
 System.out.println("65");
 }
 else if(i==0)
 {
 System.out.println("0");
 }
 else
 {
 System.out.println("garbage");
 }
}
```

A: output "0"

B: NullPointerException

**C: Compilation error**

D: output "garbage"

6. Given

```
public class Trial
{static Double d;
 public static void main(String args[])
 {
 if(d==0)
 {
 System.out.println("0");
 }
 else
 {
 System.out.println("garbage");
 }
 }
}
```

**A: it will fail at runtime**

B: output 0

C: output garbage

D: compiletime error

7. Which statement is wrong?

A: Externalizable is child of Serializable

**B: String class is final hence cannot be serialized**

C: When a class implements Serializable and it is deserialized using readObject(), constructor is never invoked.

D: all the wrapper classes they implement Serializable

8. Finalize method is a method of the class

A: String

B: Exception

**C: Object**

D: None of the above

9. Which of the following can be referenced by this variable?

A: The instance variables of a class only

B: The methods of a class only

**C: The instance variables and methods of a class**

D: The class variable

10. Which statement is true about a static nested class?

A: You must have a reference to an instance of the enclosing class in order to instantiate it.

**B: It does not have access to non-static members of the enclosing class.**

C: its variables and methods must be static.

D: must extend the enclosing class.

11. Which of the following methods cause the string object referenced by s to be changed?

A: s.concat()

B: s.touppercase()

C: s.replace()

**D: None of the above**

12. Given

```
{
public static void main(String [] args)
{
 PassA p = new PassA();
 p.start();
}

void start()
{
 long [] a1 = {3,4,5};
 long [] a2 = fix(a1);
 System.out.print(a1 [0] + a1 [1] + a1 [2] + " ");
 System.out.println(a2[0] + a2[1] + a2[2]);
}
long [] fix(long [] a3)
{
 a3[1] = 7;
 return a3;
}
}
```

A: 1 2 1 5

B: 1 5 1 5

C: 3 4 5 3 7 5

**D: 3 7 5 3 7 5**

13. What is the result of the following code?

```
import java.util.*;
```

```
enum Animals
```

```
{
 DOG("woof"), CAT("meow"), FISH("burble");
 String sound;
 Animals(String s) { sound = s; }
}
```

```
public class test11 {
 static Animals a;
 public static void main(String [] args) {
 System.out.println(a.DOG.sound + " " +
 a.FISH.sound);
 }
}
```

A: Multiple compilation errors

**B: woof burble**

C: Compilation fails due to an error on line 3

14. Inner class gets access to

**A: outer class variables**

B: outer class variables only if we created outer class object in inner class.

C: inner class variables only

D: none of the above.

15. Which of the following is not a wrapper class?

**A: String**

B: Integer

C: Boolean

D: Character

16. What is the output?

```
class A
{
 int i,j;
 A()
 {
 i=1;j=2;
 }
}

public class Abc {
 public static void main(String[] args) {
 // TODO Auto-generated method stub
 A obj1=new A();
 A obj2=new A();
 System.out.println(obj1.equals(obj2));
 }
}
```



}

- a. true
- b. false**
- c. compiler error
- d. runtime error

17. Which of the following is not abstract?

- a) **Thread**                      b) Collection                      c) ArrayList                      d) List

18. To provide access to members of the class to another class in different package which access specifier is used?

- a) **Public**                      b) protected                      c) private                      d) no modifier

19. Which of these methods is rounding function of Math class?

- a) max()                      b) min()                      c) **abs()**                      d) all of the above

20. In java System.out is an object of type \_\_\_\_\_

- a) InputStream                      b) **PrintStream**                      c) OutputStream                      d) BufferedInputStream

21. Which of the following statement is supported by an Anonymous inner class supports?

- a. It can extend exactly one class and implement exactly one interface
- b. It can extend exactly one class and can implement multiple interfaces
- c. It can extend exactly one class or implement exactly one interface**
- d. It can implement multiple interfaces regardless of whether it also extends a class.

22. Which string instance method would return true when invoked like this:

a.method(b)

where a="BUTTERfly" and b="butterFLY"

- a) **equalsIgnoreCase()**                      b) toUpperCase()                      c) toLowerCase()                      d) equals()

23. Which of the following is an ability of Reflection API in java?

- a. Determining the state of an object
- b. Determining object validity
- c. Determining duplicate classes
- d. Determination of the class of an object**

24. What is the difference between this() and super() ?

- a. super() constructor is invoked within a method of a class while this() constructor is used within the constructor of the sub class.
- b. this() constructor is invoked outside a method of a class while super() constructor is invoked within the constructor of the sub class.
- c. this() constructor is invoked within a method of a class while super() constructor is invoked outside the constructor of the sub class.

- d. **this()** constructor is invoked within a constructor of a class while **super()** constructor is used within the constructor of the sub class.

25. What is the output of the following?

```
public class MyClass
{
 public static void main(String args[])
 {
 StringBuffer sb1=new StringBuffer("Anurag");
 StringBuffer sb2=new StringBuffer("Anurag");
 String ss1="Anurag";

 System.out.println(sb1==sb2);
 System.out.println(sb1.equals(sb2));
 System.out.println(sb1.equals(ss1));
 System.out.println("Poddar".substring(3));
 }
}
```

- a) False , true , true, dar
- b) False, true, false, ddar
- c) Compiler error
- d) **false,false, false , dar**

26. Given following code, what will happen to the output?

```
public class MyClass
{
 public static void main(String args[])
 {
 String str1="hello";
 String str2="hel";
 String str3=str2+"lo";

 if(str1==str3)
 {
 System.out.println("str1 and str3 are
==");
 }
 else
 {
 System.out.println("str1 and str3 are
not ==");
 }
 }
}
```

```
 }

 if(str1.equals(str3))
 {
 System.out.println("str1 and str3 are
equals");
 }
 else
 {
 System.out.println("str1 and str3 are
not equals");
 }
}
```

a) str1 and str3 are ==  
str1 and str3 are equals

**b) str1 and str3 are not ==  
str1 and str3 are equals**

c) str1 and str3 are ==  
str1 and str3 are not equals

d) compilation error

27. Select a wrong statement about native method.

- a. Native method can be static
- b. Native method can be abstract**
- c. Native method can be non-static
- d. Native method can be synchronized

28. Constructor is the class that does not provide information about, and access to, a single constructor of a class.

- a. True                      b. **False**

29. A class cannot be both abstract and final..

- a. True**                      b. False

30. String s1="hello"; String s2="hello"; which one will return true

- a) s1==s2                      b) s1.equals(s2)                      c) **both a and b**

31. What is the correct ordering for the import, class and package declaration when found in a single file?

- a. **package, import, class**
- b. class, import, package
- c. import, package, class
- d. package, class, import

32. When native method resolution fails we get

- a. NativeResolutionFailedException
- b. NullPointerException
- c. **UnsatisfiedLinkError**
- d. None of these

33. Select the correct statement about Functional Interface.

- a. It should not contain default or static methods
- b. **It should contain only one abstract method.**
- c. It should contain more than one abstract methods.
- d. None of these.

34. Which operation is allowed on String class

- a. +
- b. -
- c. &
- d. &&

35. Using reflection u can

- a. Access private fields
- b. Access private methods
- c. **Both a and b**
- d. None

36. JRE contains

- A. Jvm
- B. jars
- C. dlls
- D. **all of the above**

37. Main() function is invoked by

- A. Programmer
- B. class\_loader
- C. **jvm**
- D. none of the above

38. Compiler which converts bytecode to native code is

- A. **Jit\_compiler**
- B. javac\_compiler
- C. byte\_compiler
- D. none of the above

39. Data types in java are

- A. Primitive\_type
- B. reference\_type
- C. **both a and b**
- D. none of these

40. What is the correct order?

- A. Linking\_loading\_initializing
- B. **loading\_linking\_initializing**
- C. initializing\_loading\_linking
- D. loading\_initializing\_linking

41. Address of next executing instruction is stored inside

- A. method\_area
- B. stack
- C. heap
- D. **PC\_Register**

42. Method area stores information about

- A. Class\_bytecode
- B. static\_variables
- C. method\_names
- D. **all of the above**

43. In java objects are created on

- A. Stack
- B. **heap**
- C. both A & B
- D. none of the above

44. Which of the following statements is true?

- A. Main is public
- B. Main is static
- C. Main accepts String[]
- D. **All of the above**

45. According to the new version of java, along with byte,short,int ,char following type is also allowed

- A. Double
- B. float
- C. **String**
- D. none of the above

46. By-default value for the Reference type is:

- A. false
- B. 0
- C. **null**
- D. none of these

47. Java does not support
- A. pointers
  - B. friend\_keyword
  - C. multiple\_inheritance
  - D. **all of the above**
48. In java by default member functions are
- A. static
  - B. **virtual**
  - C. final
  - D. all of the above
49. Just before object gets garbage collected following method is called
- A. **finalize()**
  - B. gc()
  - C. main()
  - D. none of the above
50. In java the rule is
- A. member variable must be initialized before use
  - B. **local variable must be initialized before use**
  - C. both a and b
  - D. none of these
51. What will happen if static modifier is removed from the signature of the main method?
- A - Compilation Error.
  - B - **RunTime Error: NoSuchMethodError.**
  - C - Program will compile and run without any output.
  - D - Program will compile and run to show the required output.
52. Under what conditions is an object's finalize() method invoked by the garbage collector?
- A - **When it detects that the object has become unreachable.**
  - B - As soon as object is set as null.
  - C - At fixed intervalm it checks for null value.
  - D - None of the above.
53. Can constructor be inherited?
- A - True.
  - B - **False.**
54. Under what conditions is an object's finalize() method invoked by the garbage collector?
- A - **Just before object gets garbage collected.**
  - B - As soon as object is set as null.
  - C - At fixed intervalm it checks for null value.
  - D - None of the above.
55. What is the output?
- ```
public class test10
```

```
{  
  
    static void call(int x)  
    {  
        x+=2;  
    }  
    public static void main(String args[])  
    {  
        int num=0;  
        call(num++);  
        System.out.println(num);  
    }  
}
```

- A. 1
- B. 2
- C. 3
- D. 0

56. Which of the following is the correct syntax for suggesting the JVM performs garbage collection.

- A. System.free ();
- B. System.setGarbageCollection () ;
- C. System.out.get () ;
- D. **System.gc ();**

57. Which of the following is not primitive data type?

- A. int
- B. boolean
- C. **String**
- D. float

58. Static member scope is _____

- A. **They are created when the class is loaded at runtime.**
- B. They are created when main get called.
- C. They are created when class object get created.
- D. They are created when class get modified.

59. What will be the result of attempting to compile and run the following code?

```
public class test3  
{  
    static int a;  
    int b;  
    public test3()  
    {  
        int c;  
        c=a;  
    }  
}
```

```
        a++;  
        b+=c;  
    }  
    public static void main(String args[])  
    {  
        new test3();  
    }  
}
```

Select the one correct answer

- A. The code will fail to compile since the constructor is trying to access static members
- B. The code will fail to compile since the constructor is trying to use static field "a " before it has been initialized.
- C. The code will fail to compile since the constructor is trying to use static field "b " before it has been initialized.
- D. The code will fail to compile since the constructor is trying to use static field "c " before it has been initialized.
- E. **The code will compile and run without any problems.**

SERVLET

1. Select the correct statement

A: ServletConfig is not available inside constructor

B: servlet gets instantiated every time a new request comes.

C: programmer has to override "service()" method of parent class

D: GenericServlet is a concrete class

2. One of the following is not a Parameter

A: init

B: request

C: session

D: context

3. In order to retrieve existing session only [not to create new] which function should be used ?

A: getSession(true)

B: getSession()

C: getSession(false)

D: all of the above

4. Select wrong statement

A: forward is faster than redirect

B: redirect is only within the same context

C: if client does not accept cookies, URLRewriting is the best option you have.

D: cookies are internally used inside session mechanism.

5. Cookies are always added to?

- e. Request
- f. **Response**
- g. Session
- r. Application

5. What is the difference between doing an *include* or a *forward* with a RequestDispatcher?

- A. **The forward method transfers control to the designated resource, while the include method invokes the designated resource, substitutes its output dynamically in the display, and returns control to the calling page.**
- B. The two methods provide the same functionality, but with different levels of persistence
- C. The forward method is deprecated as of JSP 1.1 and the include method should be used in order to substitute portions of a dynamic display at runtime.
- D. The include method transfers control to a dynamic resource, while the forward method allows for dynamic substitution of another JPS pages output, returning control to the calling resource.

6. Select the wrong statement

- A: ServletConfig is not available inside constructor
- B: servlet gets instantiated every time a new request comes.**
- C: programmer need not override "service()" method of parent class, instead he can override required doXXX() methods.
- D: GenericServlet is abstract class

7. One of the following is not an attribute

- A: init**
- B: request
- C: session
- D: context

8. Which of the following variables are thread-safe?

- a. session-scoped
- b. application-scoped
- c. servlet instance
- d. request-scoped**

9. execute() for any DML or DDL return Boolean

10. Technique for maintaining anonymous user transaction on web

- a) cookies b) URL Rewriting c) session **d) all of the above**

11. what is recommended for servlet Thread safety

1. Your servlet service() method should not access any member variables, unless these member variables are thread safe themselves.
2. Your servlet service() should not reassign member variables, as this may affect other threads executing inside the service() method. If you really, really need to reassign a member variable, make sure this is done inside a synchronized block.
3. Rule 1 and 2 also counts for static variables.
4. Local variables are always thread safe. Keep in mind though, that the object a local variable points to, may not be so. If the object was instantiated inside the method, and never escapes, there will be no problem. On the other hand, a local variable pointing to some shared object, may still cause problems. Just because you assign a shared object to a local reference, does not mean that object automatically becomes thread safe.

The request and response objects are of course thread safe to use. A new instance of these are created for every request into your servlet, and thus for every thread executing in your servlet.

12. Which of the following method is not used to retrieve request parameters?

- A: getParameterNames()
- B: getParameter()
- C: getParameterValue()**
- D: None of the Above

13. How to package and deploy web application in j2ee?

- A: jar
- B: ear
- C: war**
- D: rar

14. Depending upon the events on servlet we can specify the following levels_____.

- A: Request level events
- B: ServletContext level events
- C: Only A
- D: Both A and B**

15. An _____ object always represents a client's HTTP request

- A: ClientRequest
- B: HttpServlet
- C: HttpServletRequest**
- D: javax.servlet.http.HttpSession

16. You want to use URL rewriting to support client browsers, which do not use cookies. Which method will you use to attach the session id to a URL that is to be used for the sendRedirect() method of the HttpServletResponse?

A: encodeURL

B: encodeRedirectURL

C: encodeSessionURL

D: encodeSessionRedirectURL

17. Given that the service() method of a typical servlet is multithreaded, which one of the following issues does NOT need to be addressed in a servlet's implementation?

- a. Concurrent access to shared resources
- b. Concurrent access to local variables**
- c. Concurrent access to static variables
- d. Concurrent access to instance variables

18. _____ is the object used for reading "init" or "config" parameters.

- A) ServletContext **b) ServletConfig** c) RequestDispatcher

19. Request parameters are set

- a) in DD b) using setParameter **c) implicitly**

20. In case of redirect request is same. **False**

21. Following methods are the part of servlet lifecycle. [Choose 3 correct answers]

- A) Start **b) init** c) stop **d) service** **e) destroy**

22. In case of servlet, for each request

- a) each thread is created** b) each process is created c) none of them

23. Container adds cookies inside

- a) Response** b) request c) context d) config

24. ____ **web.xml** _____ file is called as Deployment Descriptor.

23. One of the following is not a servlet attribute

- a) config** b) context c) request d) session

24. If the request is get

- a) service() calls "doPost()" method **b) service() calls "doGet()"**
c) container directly calls "doGet()" d) doGet() calls "service()"

25. One of the followings is must for a servlet class

- a) it must override "service()" method b) it must have parameterized constructor
c) it must have public no-arg constructor d) it must implement Serializable interface.

26. Redirect can go beyond the context . **True**

27. HttpSession session =request.getSession()

- a) Can create a new session only b) can retrieve an existing session only
c) Can either create a new or retrieve an existing session , depends upon session availability.

28. Following interface is used to either forward or include other web resource.

- A) Servlet **b) RequestDispatcher** c) ServletConfig d) HttpServletResponse

29. Besides the cookie object which other object is also required to create a cookie on the browser?

- a. Request
b. Response
c. Session
d. Application

30. The limitations of Cookies are

- a. 20 cookies per site each of max 4Kb**
b. 4 cookies per site each of max 20Kb
c. Any no of cookies per site each of max 4Kb
d. No limitation

31. Select correct statement

A: forward is faster than redirect

B: redirect has always a same request.

C: if client does not accept cookies , session mechanism is not possible at all.

D: cookies can store java objects.

32. Your web application named bank uses "WelcomeServlet". Where will u store "WelcomeServlet.class"?

- a. bank/WEB-INF/classes**
b. bank/WEB-INF/lib/classes
c. root/WEB-INF/lib
d. bank/WebContent/lib/classes

33. What method can be used to retrieve value of the request parameter being sent as a part of the request by client?

- a. Use the method "HttpServletRequest.getParameter(string name) which will return String form.**
b. Use the method "HttpServletRequest.getParameterValues() which will return array of String values.
c. Use the method "HttpServletResponse.getValues()" which will return array of string values.
d. There is no direct support in servlet api.

34. Which of these true about deployment descriptor?
- a. The order of elements in deployment descriptor is important. The elements must follow a specific order.
 - b. The servlet-mapping element if present must be defined within “servlet” element.
 - c. The elements of deployment descriptor are not case insensitive.**
 - d. The web-app element must include the servlet element.
35. If the HTTP error 500 is generated by your servlet, you do not want to show the “Internal Server Error” page to the client. Instead, you want a custom error page to be displayed. What is the best way to accomplish this?
- a. Forward the user to the error page using `HttpServletResponse.sendRedirect()` method.
 - b. Forward the user to the error page using `RequestDispatcher.forward()` method.
 - c. Specify the mapping of the error-code 500 and the error-page in the deployment descriptor.**
 - d. It is not possible to accomplish this.
36. Which of the following should not be used to share data between servlets in a distributed web application?
- a. Attributes of ServletContext**
 - b. Enterprise JavaBeans
 - c. Attributes of HttpSession
 - d. Database
37. You want to use URL Rewriting to support client browsers which do not use cookies. Which method will you use to attach the session id to a url that is to be used for the “sendRedirect()” method of the `HttpServletResponse`?
- a) `encodeURL` b) **`encodeRedirectURL`** c) `encodeSessionURL` d) `encodeSessionRedirectURL`
38. What is the main difference between servlet and cgi?
- a. Servlets are thread based , cgi are process based**
 - b. Servlets execute slower compare to cgi
 - c. Servlets have no platform specific API, whereas cgi has
 - d. All of the above
39. A user types the URL `http://www.cdac.in` which Http request gets generated?
- a. GET method**
 - b. POST method
 - c. HEAD method
 - d. PUT method
40. Which object is used to forward the request processing from one servlet to another?
- a. `ServletContext`
 - b. `ServletConfig`
 - c. RequestDispatcher**
 - d. `ResponseDispatcher`

41. The method `getWriter()` returns an object of type `PrintWriter`. This class has `println()` method to generate output. Which of these classes define the `getWriter` method? Select the one correct answer.
- a. `HttpServletRequest`
 - b. `HttpServletResponse`**
 - c. `ServletContext`
 - d. `ServletConfig`
42. Which of the following statements is correct?
- a. The response from the dedicated server to a HEAD request consists of status line, content type and the document.
 - b. The response from the server to a GET request does not contain a document.
 - c. The `setStatus` method defined in the `HttpServletRequest` class takes `int` as an argument and sets the status of Http response.
 - d. The `HttpServletResponse` defines constants like `SC_NOT_FOUND` that may be used as a parameter to `setStatus` method.**
43. Which of the following method is not used to retrieve request parameters
- A: `getParameterNames()`
B: `getParameter()`
C: `getParameterValue()`
D: None of the Above
44. Redirect cannot go beyond context. **False**
45. Lifecycle methods of filters are [select 3 correct answers]
- a) init** b) service **c) destroy** **d) doFilter** e) intercept
46. If the request method is "get", which method is called
- A. `DoGet`**
 - B. `DoPut`
 - C. `DoPost`
 - D. `DoDelete`
47. Select the correct statement.
- A. Servlet is multithreaded
 - B. `HttpServletRequest` extends `GenericServlet`
 - C. CGI creates different process for each request
 - D. all of the above**
48. In order to make Servlet entry inside DD, we need to mention _____.
- A. Servlet-Class

- B. Servlet-name
- C. URL-pattern
- D. **All of these**

49. Request and response are created for each thread

A. False

B. **True**

50. One of the following is one per servlet

- A. Request
- B. **Config**
- C. Context
- D. All of these

51. Life cycle method in case of servlet is

- A. Destroy
- B. Service
- C. Init
- D. **all of these**

52. Service method is declared as throws

- A. ServletException
- B. IOException
- C. **both a and b**
- D. none of these

53. DD is _____.

- A. Config.xml
- B. None of these
- C. Webconfig.xml
- D. **Web.xml**

54. getWriter() method is invoked on _____.

- A. Request
- B. Config
- C. **Response**
- D. context

55. HttpServlet class is abstract class

A. False

B. **True**

56. Request Dispatcher reference can be used for forward as well as include

A. False

B. **True**

57. One of the following is invoked for every request

- A. Constructor
- B. **Service**
- C. Destroy
- D. Init

58. Which one is one per context?

- A. ServletContext**
- B. ServletConfig
- C. Both
- D. None

59. ServletConfig is available inside constructor

A. **False**

B. True

60. Redirect can go outside container _____.

A. False

B. **True**

61. init() calls init(ServletConfig)

A. **False**

B. True

62. Which one is one per servlet_____.

- A. ServletContext
- B. ServletConfig**
- C. Both
- D. None

63. sendRedirect() method is invoked on _____.

- A. HttpServletResponse**
- B. ServletContext
- C. ServletConfig
- D. HttpServletRequest

64. Forward cannot go outside context _____.

A. **False**

B. True

65. addCookie() method belongs to _____.

- A. config
- B. context
- C. request
- D. **response**

66. getCookies() method belongs to _____.

- A. A.ContextT
- B. request**
- C. response
- D. config

67. Forward is slow as compare to redirect

A. **False**

B. True

68. Init method has argument:

- A. InitConfig

- B. **ServletConfig**
- C. ServletContext
- D. None of these

69. In order to read config parameters which method is used _____.

- A. getParameter
- B. **getInitParameter**
- C. getServletParameter
- D. getConfigParameter

70. If u want to ensure servlet loading even before first request u need to use

- A. <start-on-load>
- B. **<load-on-startup>**
- C. <load-on-start>
- D. none of these

71. HttpServletRequest and HttpServletResponse are thread-safe

- A. False
- B. **True**

72. HttpSession gets migrated to another JVM if required

- A. **True**
- B. False

73. Cookies can store java objects

- A. True
- B. **False**

74. In order to get existing session only we need to use _____.

- A. getSession()
- B. getSession(true)
- C. getExistingSession()
- D. **getSession(false)**

SOCKET PROGRAMMING

1. Which of the following class allows Tcp Server to wait for client on a particular port?

- A: InetAddress
- B: ServerSocket**
- C: Socket
- D: none of the above

2. One of the following port range is valid for Network programming in java

- A: 1 to 65535
- B: 1023 to 65535
- C: 1024 to 65535**
- D: 0 to 1023

3. Which one is used to send packet over the network in case of UDP?

A: DatagramPacket

B: Socket

C: DatagramServer

D: DatagramSocket

4. Which of the following is Application level protocol?

A: FTP

B: HTTP

C: JRMP

D: all of the above

5. A _____ is an endpoint for communication between two machines.

a) ServerSocket

b) Socket

c) DatagramSocket

d) DatagramPacket

6. Which of the following class allows UDP Server to wait for client on a particular port?

A: InetAddress

B: DatagramSocket

C: DatagramPacket

D: none of the above

7. One of the following class is used to represent IP address of a machine.

A: IPAddress

B: InetAddress

C: InternetAddress

D: InternetPacketAddress

8. Which method is used to wait for client to get connected in TCP?

A: accept

B: receive

C: wait

D: socketWait

9. Which of the following is Application level protocol?

A: TCP

B: HTTP

C: UDP

D: all of the above

10. The class which is used to send the packet in case of UDP is
- A. Socket
 - B. UDP socket
 - C. DatagramPacket
 - D. **DatagramSocket**
11. The class which represents IP address of machine is
- A. InetAddress
 - B. IPAddress
 - C. InetAddress
 - D. none of the above
12. Which is Application layer
- A. HTTP
 - B. FTP
 - C. SMTP
 - D. **all of the above**
13. _____ method is used to wait for client request in UDP
- A. Wait
 - B. **receive**
 - C. accept
 - D. none of these
14. _____ method is used to wait for client request in TCP
- A. Wait
 - B. receive
 - C. **accept**
 - D. none of these
15. If we want to pass an object over network it should implement
- A. Runnable
 - B. **Serializable**
 - C. Cloneable
 - D. none of these
16. _____ class is used to make server wait for client request in TCP.
- A. Socket
 - B. **ServerSocket**
 - C. SocketInputStream
 - D. none of these
17. Valid range of port number for a java application is
- A. 0 to 65535
 - B. 1 to 65535
 - C. **1024 to 65535**

D. none of these

18. Marshalling is

- A. Converting packets into data
- B. **converting data into packets**
- C. converting bytes into character

19. TCP is reliable

A. True

B. false

STRUTS

1. Following is the main controller in struts2 framework

A: FilterDispatcher

B: ActionServlet

C: FilterServlet

D: FilterController

2. One of the following is not a result type

A: dispatcher

B: chain

C: response

D: redirect

3. Action objects are stored in

A: value stack

B: heap

C: file

D: none of these

4. In action tag, when method is not specified, which method is by default considered?

A: run

B: actionexecute

C: execute

D: perform

5. One of the following is the map in OGNL to retrieve any type of attribute

A: request

B: application

C: page

D: attr

6. Which is the interface used if you want to access "request" object in your action class?

A: ServletRequest

B: HttpServletRequest

C: ServletRequestAware

D: RequestAware

7. Action methods must return

A: String

B: int

C: boolean

D: it can return any type

8. One of the following is a result type

A: dispatcher

B: error

C: response

D: request

9. struts uses _____ for multiple language support

a) resource

b) resource bundle

c) language bundle

d) all of the above

10. EL maps

| | |
|------------------|---|
| pageScope | Scoped variables from page scope |
| requestScope | Scoped variables from request scope |
| sessionScope | Scoped variables from session scope |
| applicationScope | Scoped variables from application scope |
| param | Request parameters as strings |
| paramValues | Request parameters as collections of strings |
| header | HTTP request headers as strings |
| headerValues | HTTP request headers as collections of strings |
| initParam | Context-initialization parameters |
| cookie | Cookie values |
| pageContext | The JSP PageContext object for the current page |

11. Controller component in struts 2 is

A: filter

B: servlet

C: jsp

D: None of the above.

12. To read value from the value stack we need to use

A: SQL

B: PU/SQL

C: OGNL

D: None of the above.

13. Interceptor does _____&_____processing of request and invokes the action classes.

A: pre, post

B: before, after

C: get, post

D: None of the above

14. Which interceptor is used for internationalization in struts?

A: struts.custom.i17n.resources

B: struts.custom.i18n

C: struts.custom.i18n.resources

D: struts.i18n.custom.resources

15. Default Interceptors of struts 2 are

A: ConversionError, ModelDriven, jsonValidationparams

B: ModelDriven, jsonvalidation, ParameterDriven

C: ConversionError, ParameterDriven, roleDriven

D: ConversionError, roleDriven, jsonValidation, params

16. Struts2 application flow is defined in a configuration file called _____

a) web.xml

b) struts.xml

c) struts-config.xml

17. The controller in struts2 is _____

a) ActionServlet

b) HttpServlet

c) FilterDispatcher

18. OGNL stands for __Object Graph Navigation Language_____

19. _____ is the interceptor stack available by default to every struts application.

A) execAndWait

b) validateStack

c) defaultStack

20. By default result name is _____

a) success

b) fail

c) retry

d) abort

21. In action tag, when method is not specified _____ is the by default method considered.

A) run

b) execute

c) actionExecute

d) perform

22. We can forward the request from one action class to another. **True**

23. Tiles definitions are mentioned inside _____ file.

A) Struts.xml

b) tiles.xml

c) web.xml

d) tiles-def.xml

24. Interceptors are similar to _____ in concept.

A) Filters

b) servlets

c) jsps

d) pojos

25. Action method has to return

- a) **String** b) int c) Boolean d) it can return any type

26. Which method is necessary for Action class?

- a) Valuator () b) reset() c) findForward() **d) execute()**

27. What is the return type of validate() method in struts?

- a) String b) **void** c) int d) Object

28. Can I add user defined data in valueStack in struts?

- a. Depends
b. Yes always
c. Not possible
d. None of the above

29. _____ is the interface used if you want to access "request" object in your action class.

- A) ServletRequest b) HttpServletRequest **c) ServletRequestAware** d) RequestAware

30. If the action name is "AddAction", the validation file name should be _____

- a) AddAction-validation.xml**
b) AddAction-validator.xml
c) AddAction_validation.xml

SPRING

1. Spring controller can be defined using annotation

- a) @Bean
b) @Component
c) @Controller
d) none of the above

2. Mapping of request in Spring MVC can be defined using annotation

- a) @RequestMapping**
b) @Action
c) @RequestProcessing
d) @Request

3. The advice functionality takes place after the advised method completes, regardless of the outcome:

- a) after-returning
b) After
c) after-throwing
d) None of the above

4. In Spring framework by default the scope of bean is
- A. Prototype
 - B. Request
 - C. Session
 - D. Singleton**
5. Which class is used to check spring lifecycle?
- a. BeanFactory
 - b. ClassPathApplicationContext**
 - c. AbstractApplicationContext
 - d. All of the above
6. Front Controller in Spring MVC is
- a) DispatcherServlet**
 - b) ActionServlet
 - c) FacesServlet
 - d) none of the above
7. Join points can be method calls, constructor invocations, exception handlers, or other points in the execution of a program.
- a) true**
 - b) false
8. When a lazy-initialized bean is a dependency of a singleton bean that is not lazy-initialized, the ApplicationContext creates the lazy-initialized bean at startup
- a) True**
 - b) False
9. Spring DAO framework converts checked exception into unchecked exception. The name of unchecked exception is _____
- a) RuntimeException
 - b) SQLException
 - c) DataAccessException**
 - d) none of the above
10. The class which encapsulates traditional jdbc code is
- a) DataTemplate
 - b) JdbcTemplate**
 - c) SqlTemplate
 - d) none of the above
11. One of the following helps controller in resolving a particular view component.

- a) ModelAndView
- b) HandlerMapping
- c) ViewResolver**
- d) DispatcherServlet

12. What are the types of Dependency Injection Spring supports?

- a) Setter injection
- b) Constructor injection
- c) Both a and b**
- d) none of the above

13. The process of applying aspects to a target object to create a new proxy object is called as _____

- a) Coupling
- b) Weaving**
- c) Injecting
- d) None of the above

14. Annotation to force dependency injection is _____

- a) @Injection
- b) @Autowired**
- c) @Wired
- d) None of the above

15. In order to use @Component annotation for a bean u need to add following tag in Spring bean configuration file.

- a) <context:component-scan>**
- b) <auto:component-scan>
- c) <context:bean-scan>
- d) None of the above

16. What is the Key strategy of spring framework?

- A: Application development in POJO style.
- B: Loose coupling through DI.
- C: Declarative programming using AOP
- D: All of the above.**

SWING

1. Select correct statement from the following

- A: Invisible components are required in BorderLayout**
- B: flowlayout is the default layout for JApplet
- C: the default lookandfeel for swing components is MotifLookAndFeel.
- D: swing does not have DelegationEvent model.

2. How will you create Dual Application in Swing?

- A: extend JFrame, inside main instantiate JApplet
- B: extend JApplet, inside main instantiate JFrame**
- C: extend JFrame as well as JApplet
- D: none of these