**MCQ Java Set-4**

1. The \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ object specifies the atomic unit of work. It is optional.
   1. Session
   2. SessionFactory
   3. Transaction
   4. Query

Answer: Option C

1. \_\_\_\_\_\_\_\_\_\_\_\_is the sub-element of id. It is used to generate the primary key.
   1. **generator**
   2. **property**
   3. **class**
   4. **column**

Answer: Option A

1. What does Class.forName(“myreflection.Foo”).getInstance() return?

* [**A.**](javascript:%20void(0)) An array of Foo objects
* [**B.**](javascript:%20void(0)) class object of Foo
* [**C.**](javascript:%20void(0)) Calls the getInstance() method of Foo class
* [**D.**](javascript:%20void(0)) Foo object

Answer: Option D

1. What are the drawbacks of Java Reflection?

* [**A.**](javascript:%20void(0)) Performance Overhead
* [**B.**](javascript:%20void(0)) Exposure of Internals
* [**C.**](javascript:%20void(0)) Both A and B
* [**D.**](javascript:%20void(0)) Not Synchronized

Answer: Option C

1. How method is to be invoked on Unknown object?

* [**A.**](javascript:%20void(0)) obj.getClass().getDeclaredMethod()
* [**B.**](javascript:%20void(0)) obj.getClass().getDeclaredField()
* [**C.**](javascript:%20void(0)) obj.getClass().getMethod()
* [**D.**](javascript:%20void(0)) obj.getClass().getObject()

Answer: Option C

1. What are the advantages of using reflection?

* [**A.**](javascript:%20void(0)) Extensibility Features
* [**B.**](javascript:%20void(0)) Debugging and testing tools
* [**C.**](javascript:%20void(0)) Both A and B
* [**D.**](javascript:%20void(0)) None of these

Answer: Option C

Threads MCQ

1. **What will be the output of the program?**

|  |
| --- |
| class Test extends Thread {  public      void run()      {          System.out.println("Run");      }  } class Myclass {  public      static void main(String[] args)      {          Test t = new Test();          t.run();      }  } |

Options:  
1. One thread created  
2. Two thread created  
3. Depend upon system  
4. No thread created

Output:

The answer is option (1)

1. **What is NORM\_PRIORITY value in thread priority in Java.**
2. 10
3. 12
4. 5
5. 8

Answer: C

##### Which statements is/are correct

1. On calling Thread wait () method a new thread get created.
2. Thread start () method call run () method internally
3. The run() is a method Thread class.
4. Runnable interface has only one method.

Answer: B, D

1. What are the two types of synchronization in java:
2. Process Synchronization and Thread Synchronization
3. Class Synchronization and Method Synchronization
4. Field Synchronization and Process Synchronization
5. Thread Synchronization and Method Synchronization

Answer: A

##### Which method is used to create a daemon thread?

1. setDaemon(boolean value)
2. setThread(boolean value)
3. createDaemon(boolean value)
4. makeDeamon(boolean value);

Answer : A

# MCQ – Java Exceptions

##### What exception can occur in the below java program?

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | public class MyException {    public static void main(String[] args) {    try {    int a[] = { 5, 10, 15, 20 };    System.out.println("Element :" + a[4]);  }  finally{}    }    } |

1. ArrayIndexOutOfBoundsException
2. ArithmeticException
3. NullPointerException
4. No Exception in the program

Answer: A

##### If an exception is thrown from try block in java program then which block handle the exception?

* catch
* finally
* final
* finalize

Answer: A

##### Which of the following option is not an inbuild exception class in Java?

1. IOException
2. LogicalException
3. ArrayIndexOutOfBoundsException
4. ArithmeticException

Answer: B

1. Which of the following statement about inheritance is NOT true:
2. **Subclass** is a class which inherits the other class. It is also called a derived class, extended class, or child class.
3. **Superclass** is the class from where a subclass inherits the features. It is also called a base class or a parent class.
4. The **extends keyword** indicates that you are making a new class that derives from an existing class. The meaning of "extends" is to increase the functionality.
5. **Multiple inheritance** is supported in Java through class.

Answer: D

##### In below java program, which exception will occur(Assume “test.txt” is there)”?

public static void main(String[] args) {

     FileReader file = new FileReader("test.txt");

    }

1. NullPointerException at compile time
2. NullPointerException at run time
3. FileNotFoundException at compile time
4. FileNotFoundException at runtime

Answer: C

1. **java interface can contain ————**
2. public static Final Variables only
3. public Abstract methods
4. Abstract methods(unimplemented) and implemented methods both
5. public static Final Variables and abstract methods both

Answer: D

1. **Zara want to add additional role to the existing class to inherits an interface she should use which keyword?**
2. extends
3. implements
4. inherits
5. implement

Answer: B

1. What is output of the below java program?

interface IShape {

void f();

}

class Circle implements IShape {

public void f() {

System.out.println("Interface");

}

public void c() {

System.out.println("class");

}

}

public class Main {

public static void main(String[] args) {

IShape obj = new Circle();

obj.f();

}

}

1. Interface
2. Class
3. Compiler error
4. Runtime error

Answer: A

1. **Java interface is used to**
2. Implement behaviour of multiple inheritance
3. Achieve abstraction
4. achieve loos coupling
5. All of the above

Answer: D

##### \_\_\_\_\_\_\_\_\_\_\_\_ provides run time environment for java byte code to be executed

1. JDK
2. JRE
3. JVM
4. JAVAC

Answer: C

##### Justin was learning java fundamentals and according to you how you will tell Justin what Byte code is:

1. Machine-dependent
2. Machine-instruction
3. Machine- independent
4. HashCode

Answer: C

##### On successful compilation of a java source code \_\_\_\_\_\_\_\_\_\_\_\_\_ is generated

1. HashCode
2. Bytecode
3. Octal code
4. Hexadecimal code

Answer: B

##### Java does not support \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_?

1. Inheritance
2. Multiple inheritance for classes
3. multiple inheritance of interfaces
4. compile time polymorphism

Answer: B

##### Runtime polymorphism feature in java is

1. method overriding
2. method overloading
3. constructor overloading
4. operator overloading

Answer: A

##### If I want to have common functions in a class and want to defer implementations of some other functions to derived classes, then we need to use

1. An interface
2. An abstract class
3. A static inner class
4. Anonymous class

Answer: B

1. **Arya wants to achieve Loose coupling in her java programs which of the following option should she inculcate in her program to achieve so.**
2. Using interface
3. Encapsulating an object of another class
4. Using Inheritance
5. Inner classes

Answer: A

##### Raj wants to expose only necessary information to clients ( main programs, classes) is known as

1. Abstraction
2. Encapsulation
3. Data hiding
4. Hiding complexity

Answer: A

1. **What is the output of following Java Program?**

|  |
| --- |
| import java.util.ArrayList;  class Demo {  public void show()      {          ArrayList<Integer> list = new ArrayList<Integer>();          list.add(4);          list.add(7);          list.add(1);          for (int number : list) {              System.out.print(number + " ");          }      }  } public class Main {  public static void main(String[] args)      {          Demo demo = new Demo();          demo.show();      }  } |

A. Compilation Error  
B. 4 7 1  
C. 1 4 7  
D. None

Answer : B.

1. **What is the output of following Java Program?**

|  |
| --- |
| import java.util.ArrayList;  class Demo {  public void show()      {          ArrayList<String> list = new ArrayList<String>();          System.out.print(list.get(0));      }  } public class Main {  public static void main(String[] args)      {          Demo demo = new Demo();          demo.show();      }  } |

A ArrayException  
B. IndexOutOfBoundException  
C. null

D NullPointerException

Answer : B

1. **What is the output of this question?**

|  |
| --- |
| class TestArrays {  public static void main(String[] args)      {          int a[] = new int[10]; // line 1          int[] a11 = new int[]; // line 2      }  } |

**Option**  
A) Error at line 1  
B) Error at line 2  
C) Run successfully  
D) Runtime Exception

Answer: B

1. JDBC API uses JDBC drivers to connect with the database. There are four types of JDBC drivers. Which of the following is not JDBC driver.
2. JDBC-ODBC Bridge Driver,
3. Native Driver,
4. Network Protocol Driver,
5. Native Network Driver

Answer: D

1. The **java.sql** package contains classes and interfaces for JDBC API. Which of the following in not an interfaces of JDBC API:
2. DriverManager interface
3. CallableStatement interface
4. ResultSet interface
5. ResultSetMetaData interface

Answer: A

1. We can use JDBC API to handle database using Java program and can perform the following activities:
2. Connect to the database
3. Execute queries and update statements to the database
4. Retrieve the result received from the database.
5. All the Above.

Answer: D

1. Each time you create a string literal, the JVM checks the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ first. If the string already exists in the pool, a reference to the pooled instance is returned. If the string doesn't exist in the pool, a new string instance is created and placed in the pool.
2. Heap
3. Stack
4. String Pool.
5. None of the above.

Answer: C

1. What is the ouput of the following code:

**public** **class** StringExample{

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

String s1="java";//creating string by java string literal

**char** ch[]={'s','t','r','i','n','g','s'};

String s2=**new** String(ch);//converting char array to string

String s3=**new** String("example");//creating java string by new keyword

System.out.println(s1);

System.out.println(s2);

System.out.println(s3);

}}

1. java

strings

example

1. java

example

strings

1. example

java

strings

1. Error

Answer: A

1. In Java, 3 streams are created for us automatically. Which of the statement is incorrect.
2. **System.out:**standard output stream associated with standard output device.
3. **System.in:**standard input stream associated with standard input device.
4. **System.err:**standard error stream associated with standard output device.
5. **System.err:**standard error stream associated with standard input device.

Answer: D

1. **Which statement is true about the following code: (assume “**D:\\testout.txt” exists**)**

**import** java.io.FileOutputStream;

**public** **class** FileOutputStreamExample {

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

**try**{

             FileOutputStream fout=**new** FileOutputStream("D:\\testout.txt");

             fout.write(65);

             fout.close();

             System.out.println("success...");

            }**catch**(Exception e){System.out.println(e);}

      }

}

1. The content of a text file **testout.txt** is set with the data **A**.
2. The content of a text file **testout.txt** is set with the data **65**.
3. No content written in text file **testout.txt**.
4. Runtime Error.

Answer: A

1. Serializable is a \_\_\_\_\_\_\_\_\_\_\_\_ interface (has no data member and method). It is used to "mark" Java classes so that objects of these classes may get the certain capability.
   1. Tagging
   2. Native
   3. Inner
   4. Static

Answer: A

1. The Reflection API is mainly used in:
2. IDE (Integrated Development Environment) e.g. Eclipse, MyEclipse, NetBeans etc.
3. Debugger
4. Test Tools
5. All the above

Answer: D

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_is a powerful project management tool that is based on POM (project object model). It is used for projects build, dependency and documentation.It simplifies the build process like ANT. But it is too much advanced than ANT.
2. Maven
3. Ant
4. Hibernate
5. Reflection

Answer: A

1. Maven does not simplify which of the following tasks.
2. It makes a project easy to build
3. It provides uniform build process (maven project can be shared by all the maven projects)
4. It provides project information (log document, cross referenced sources, mailing list, dependency list, unit test reports etc.)
5. Helps to achieve Dependency Injection and Aspect Oriented Programming.

Answer: D

1. Which is the correct syntax to configure the employee.hbm.xml file in hibernate.cfg.xml file?
2. <mapping file="employee.hbm.xml"/>
3. <mapping class="employee.hbm.xml"/>
4. <mapping xml="employee.hbm.xml"/>
5. <mapping resource="employee.hbm.xml"/>

Answer: D

#### In the following mapping file “**employee.hbm.xml**”

What is the “**assigned**” generator type means :

<?xml version='1.0' encoding='UTF-8'?>

<!DOCTYPE hibernate-mapping PUBLIC

 "-//Hibernate/Hibernate Mapping DTD 5.3//EN"

 "http://hibernate.sourceforge.net/hibernate-mapping-5.3.dtd">

 <hibernate-mapping>

  <**class** name="com.synergetics.hibernate.Employee" table="emp1000">

    <id name="id">

     <generator **class**="assigned"></generator>

    </id>

    <property name="firstName"></property>

    <property name="lastName"></property>

  </**class**>

 </hibernate-mapping>

1. It generates the unique id only if no other process is inserting data into this table.
2. It uses high and low algorithm to generate the id of type short, int and long.
3. It uses identity, sequence or hilo depending on the database vendor.
4. It returns same id set by the programmer to hibernate and hibernate will store an object with that Id in database. This is the default behavior if you do not specify a <**generator**> element.

Answer: D

1. The \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ property of hibernate.cfg.xml file, specifies the type of database used in hibernate so that hibernate generate appropriate type of SQL statements.
2. <property name="dialect">
3. <property name="showsql">
4. <property name=" hbm2ddl.auto">
5. <property name=" connection.url ">

Answer: A

1. The \_\_\_\_\_\_\_\_\_\_\_ **keyword** in Java is used for memory management mainly. We can apply this keyword with variables, methods, blocks and nested class. This keyword belongs to the class than an instance of the class.
2. **abstract**
3. **native**
4. **static**
5. **final**

Answer: C

1. \_\_\_\_\_\_\_\_\_\_ method is inherited but you cannot override it.
2. **abstract**
3. **final**
4. **native**
5. **static**

Answer: B

1. **Java \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ interface** is used to order the objects of a user-defined class.This interface is found in java.util package and contains 2 methods compare(Object obj1,Object obj2) and equals(Object element).
2. **Collection**
3. **Comparator**
4. **Comparable**
5. **SortedSet**

Answer: B

1. Java \_\_\_\_\_\_\_\_ class is a red-black tree based implementation. It provides an efficient means of storing key-value pairs in sorted order.
2. HashMap
3. LinkedHashMap
4. TreeMap
5. TreeSet

Answer: C