Autoboxing

1 – JUnits

1. JUnits are used for which type of testing?

a) Unit Testing

b) Integration Testing

c) System Testing

d) Blackbox Testing

Answer: a

Explanation: JUnit is a testing framework for unit testing. It uses java as a programming platform. It is managed by junit.org community.

2. Which of the below statement about JUnit is false?

a) It is an open source framework

b) It provides an annotation to identify test methods

c) It provides test runners for running test

d) They cannot be run automatically

Answer: d

Explanation: JUnits test can be run automatically and they check their own results and provide immediate feedback.

3. Which of the below is an incorrect annotation with respect to JUnits?

a) @Test

b) @BeforeClass

c) @Junit

d) @AfterEach

Answer: c

Explanation: @Test is used to annotate method under test, @BeforeEach and @AfterEach are called before and after each method respectively. @BeforeClass and @AfterClass are called only once for each class.

4. Which of these is not a mocking framework?

a) EasyMock

b) Mockito

c) PowerMock

d) MockJava

Answer: d

Explanation: EasyMock, jMock, Mockito, Unitils Mock, PowerMock and JMockit are a various mocking framework.

5. Which method is used to verify the actual and expected results in Junits?

a) assert()

b) equals()

c) ==

d) isEqual()

Answer: a

Explanation: assert method is used to compare actual and expected results in Junit. It has various implementation like assertEquals, assertArrayEquals, assertFalse, assertNotNull, etc.

6. What does assertSame() method use for assertion?

a) equals() method

b) isEqual() method

c) ==

d) compare() method

Answer: c

Explanation: == is used to compare the objects not the content. assertSame() method compares to check if actual and expected are the same objects. It does not compare their content.

7. How to let junits know that they need to be run using PowerMock?

a) @PowerMock

b) @RunWith(PowerMock)

c) @RunWith(Junits)

d) @RunWith(PowerMockRunner.class)

Answer: d

Explanation: @RunWith(PowerMockRunner.class) signifies to use PowerMock JUnit runner. Along with that @PrepareForTest(User.class) is used to declare the class being tested. mockStatic(Resource.class) is used to mock the static methods.

8. How can we simulate if then behavior in Junits?

a) if{..} else{..}

b) if(..){..} else{..}

c) Mockito.when(…).thenReturn(…);

d) Mockito.if(..).then(..);

Answer: c

Explanation: Mockito.when(mockList.size()).thenReturn(100); assertEquals(100, mockList.size()); is the usage to implement if and then behavior.

9. What is used to inject mock fields into the tested object automatically?

a) @InjectMocks

b) @Inject

c) @InjectMockObject

d) @Mock

Answer: a

Explanation: @InjectMocks annotation is used to inject mock fields into the tested object automatically.

@InjectMocks

MyDictionary dic = new MyDictionary();

10. How can junits be implemented using maven?

a)

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.8.1</version>

</dependency>

b)

<dependency>

<groupId>org.junit</groupId>

<artifactId>junit</artifactId>

<version>4.8.1</version>

</dependency>

c)

<dependency>

<groupId>mock.junit</groupId>

<artifactId>junit</artifactId>

<version>4.8.1</version>

</dependency>

d)

<dependency>

<groupId>junits</groupId>

<artifactId>junit</artifactId>

<version>4.8.1</version>

</dependency>

Answer: a

Explanation: JUnits can be used using dependency tag in maven in pom.xml. The version as desired and available in repository can be used.

2 – Java 8 Features

1. Which of the following is not introduced with Java 8?

a) Stream API

b) Serialization

c) Spliterator

d) Lambda Expression

Answer: b

Explanation: Serialization is not introduced with Java 8. It was introduced with an earlier version of Java.

2. What is the purpose of BooleanSupplier function interface?

a) represents supplier of Boolean-valued results

b) returns Boolean-valued result

c) There is no such function interface

d) returns null if Boolean is passed as argument

Answer: a

Explanation: BooleanSupplier function interface represents supplier of Boolean-valued results.

3. What is the return type of lambda expression?

a) String

b) Object

c) void

d) Function

Answer: d

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

4. Which is the new method introduced in java 8 to iterate over a collection?

a) for (String i : StringList)

b) foreach (String i : StringList)

c) StringList.forEach()

d) List.for()

Answer: c

Explanation: Traversing through forEach method of Iterable with anonymous class.

StringList.forEach(new Consumer<Integer>()

{

public void accept(Integer t)

{

}

});

//Traversing with Consumer interface implementation

MyConsumer action = new MyConsumer();

StringList.forEach(action);

}

}

5. What are the two types of Streams offered by java 8?

a) sequential and parallel

b) sequential and random

c) parallel and random

d) random and synchronized

Answer: a

Explanation: Sequential stream and parallel stream are two types of stream provided by java.

Subscribe Now: Java Newsletter | Important Subjects Newsletters

Stream<Integer> sequentialStream = myList.stream();

Stream<Integer> parallelStream = myList.parallelStream();

6. Which feature of java 8 enables us to create a work stealing thread pool using all available processors at its target?

a) workPool

b) newWorkStealingPool

c) threadPool

d) workThreadPool

Answer: b

Explanation: Executors newWorkStealingPool() method to create a work-stealing thread pool using all available processors as its target parallelism level.

7. What does Files.lines(Path path) do?

a) It reads all the files at the path specified as a String

b) It reads all the lines from a file as a Stream

c) It reads the filenames at the path specified

d) It counts the number of lines for files at the path specified

Answer: b

Explanation: Files.lines(Path path) that reads all lines from a file as a Stream.

8. What is Optional object used for?

a) Optional is used for optional runtime argument

b) Optional is used for optional spring profile

c) Optional is used to represent null with absent value

d) Optional means it’s not mandatory for method to return object

Answer: c

Explanation: Optional object is used to represent null with absent value. This class has various utility methods to facilitate code to handle values as ‘available’ or ‘not available’ instead of checking null values.

9. What is the substitute of Rhino javascript engine in Java 8?

a) Nashorn

b) V8

c) Inscript

d) Narcissus

Answer: a

Explanation: Nashorn provides 2 to 10 times faster in terms of performance, as it directly compiles the code in memory and passes the bytecode to JVM. Nashorn uses invoke dynamic feature.

10. What does SAM stand for in the context of Functional Interface?

a) Single Ambivalue Method

b) Single Abstract Method

c) Simple Active Markup

d) Simple Abstract Markup

Answer: b

Explanation: SAM Interface stands for Single Abstract Method Interface. Functional Interface is also known as SAM Interface because it contains only one abstract method.

3 – File and Directory

1. Which method is used to create a directory with fileattributes?

a) Path.create()

b) Path.createDirectory()

c) Files.createDirectory(path, fileAttributes)

d) Files.create(fileAttributes)

Answer: c

Explanation: New directory can be created using Files.createDirectory(path, fileAttribute).

2. Which method can be used to check fileAccessiblity?

a) isReadable(path)

b) isWritable(path)

c) isExecutable(path)

d) isReadable(path), isWritable(path), and isExecutable(path)

Answer: d

Explanation: File accessibilty can be checked using isReadable(Path), isWritable(Path), and isExecutable(Path).

3. How can we delete all files in a directory?

a) Files.delete(path)

b) Files.deleteDir()

c) Directory.delete()

d) Directory.delete(path)

Answer: a

Explanation: The delete(Path) method deletes the file or throws an exception if the deletion fails. If file does not exist a NoSuchFileException is thrown.

4. How to copy the file from one location to other?

a) Files.copy(source, target)

b) Path.copy(source, target)

c) source.copy(target)

d) Files.createCopy(target)

Answer: a

Explanation: Files.copy(source, target) is used to copy a file from one location to another. There are various options available like REPLACE\_EXISTING, COPY\_ATTRIBUTES and NOFOLLOW\_LINKS.

5. How can we get the size of specified file?

a) capacity(path)

b) size(path)

c) length(path)

d) Path.size()

Answer: b

Explanation: size(Path) returns the size of the specified file in bytes.

6. How to read entire file in one line using java 8?

a) Files.readAllLines()

b) Files.read()

c) Files.readFile()

d) Files.lines()

Answer: a

Explanation: Java 8 provides Files.readAllLines() which allows us to read entire file in one task. We do not need to worry about readers and writers.

7. How can we create a symbolic link to file?

a) createLink()

b) createSymLink()

c) createSymbolicLink()

d) createTempLink()

Answer: c

Explanation: createSymbolicLink() creates a symbolic link to a target.

8. How can we filter lines based on content?

a) lines.filter()

b) filter(lines)

c) lines.contains(filter)

d) lines.select()

Answer: a

Explanation: lines.filter(line -> line.contains(“===—> Loaded package”)) can be used to filter out.

9. Which jar provides FileUtils which contains methods for file operations?

a) file

b) apache commons

c) file commons

d) dir

Answer: b

Explanation: FileUtils is a part of apache commons which provides various methods for file operations like writeStringToFile.

10. Which feature of java 7 allows to not explicitly close IO resource?

a) try catch finally

b) IOException

c) AutoCloseable

d) Streams

Answer: c

Explanation: Any class that has implemented Autocloseable releases the I/O resources.

4 - Hibernate MCQ

1. Which of the following is not a core interface of Hibernate?

a) Configuration

b) Criteria

c) SessionManagement

d) Session

Answer: c

Explanation: SessionManagement is not a core interface of Hibernate. Configuration, Criteria, SessionFactory, Session, Query and Transaction are the core interfaces of Hibernate.

2. SessionFactory is a thread-safe object.

a) True

b) False

Answer: a

Explanation: SessionFactory is a thread-safe object. Multiple threads can access it simultaneously.

3. Which of the following methods returns proxy object?

a) loadDatabase()

b) getDatabase()

c) load()

d) get()

Answer: c

Explanation: load() method returns proxy object. load() method should be used if it is sure that instance exists.

4. Which of the following methods hits database always?

a) load()

b) loadDatabase()

c) getDatabase()

d) get()

Answer: d

Explanation: get() method hits database always. Also, get() method does not return proxy object.

5. Which of the following method is used inside session only?

a) merge()

b) update()

c) end()

d) kill()

Answer: b

Explanation: update() method can only be used inside session. update() should be used if session does not contain persistent object.

6. Which of the following is not a state of object in Hibernate?

a) Attached()

b) Detached()

c) Persistent()

d) Transient()

Answer: a

Explanation: Attached() is not a state of object in Hibernate. Detached(), Persistent() and Transient() are the only states in Hibernate.

7. Which of the following is not an inheritance mapping strategies?

a) Table per hierarchy

b) Table per concrete class

c) Table per subclass

d) Table per class

Answer: d

Explanation: Table per class is not an inheritance mapping strategies.

8. Which of the following is not an advantage of using Hibernate Query Language?

a) Database independent

b) Easy to write query

c) No need to learn SQL

d) Difficult to implement

Answer: d

Explanation: HQL is easy to implement. Also, to implement it HQL it is not dependent on a database platform.

9. In which file database table configuration is stored?

a) .dbm

b) .hbm

c) .ora

d) .sql

Answer: b

Explanation: Database table configuration is stored in .hbm file.

10. Which of the following is not an advantage of Hibernate Criteria API?

a) Allows to use aggregate functions

b) Cannot order the result set

c) Allows to fetch only selected columns of result

d) Can add conditions while fetching results

Answer: b

Explanation: addOrder() can be used for ordering the results.

5 – Liskov’s Principle

1. What does Liskov substitution principle specify?

a) parent class can be substituted by child class

b) child class can be substituted by parent class

c) parent class cannot be substituted by child class

d) No classes can be replaced by each other

Answer: a

Explanation: Liskov substitution principle states that Objects in a program should be replaceable with instances of their sub types without altering the correctness of that program.

2. What will be the correct option of the following Java code snippet?

interface ICust

{

}

class RegularCustomer implements ICust

{

}

class OneTimeCustomer implements ICust

{

}

a) ICust can be replaced with RegularCustomer

b) RegularCustomer can be replaced with OneTimeCustomer

c) OneTimeCustomer can be replaced with RegularCustomer

d) We can instantiate objects of ICust

Answer: a

Explanation: According to Liskov substitution principle we can replace ICust with RegularCustomer or OneTimeCustomer without affecting functionality.

3. What will be the output of the following Java code snippet?

public class Shape

{

public int area()

{

return 1;

}

}

public class Square extends Shape

{

public int area()

{

return 2;

}

}

class Main()

{

public static void main(String[] args)

{

Shape shape = new Shape();

Square square = new Square();

shape = square;

System.out.println(shape.area());

}

}

a) Compilation failure

b) Runtime failure

c) 1

d) 2

View Answer

Answer: d

Explanation: Child object can be assigned to parent variable without change in behaviour.

4. What will be the output of the following Java code snippet?

public class Shape

{

public int area()

{

return 1;

}

}

public class Rectangle extends Shape

{

public int area()

{

return 3;

}

}

class Main()

{

public static void main(String[] args)

{

Shape shape = new Shape();

Rectangle rect = new Rectangle();

shape = rect;

System.out.println(shape.area());

}

}

a) Compilation failure

b) 3

c) 1

d) 2

Answer: b

Explanation: Child object can be assigned to parent variable without change in behaviour.

5. What will be the output of the following Java code?

public class Shape

{

public int area()

{

return 1;

}

}

public class Square extends Shape

{

public int area()

{

return 2;

}

}

class Main()

{

public static void main(String[] args)

{

Shape shape = new Shape();

Square square = new Square();

square = shape;

System.out.println(square.area());

}

}

a) Compilation failure

b) 3

c) 1

d) 2

View Answer

Answer: a

Explanation: Parent object cannot be assigned to child class.

6. What will be the output of the following Java code?

public class Shape

{

public int area()

{

return 1;

}

}

public class Square extends Shape

{

public int area()

{

return 2;

}

}

public class Rectangle extends Shape

{

public int area()

{

return 3;

}

}

class Main()

{

public static void main(String[] args)

{

Shape shape = new Shape();

Square square = new Square();

Rectangle rect = new Rectangle();

rect = (Rectangle)shape;

System.out.println(square.area());

}

}

a) Compilation failure

b) 3

c) Runtime Exception

d) 2

Answer: c

Explanation: ClassCastException is thrown as we cannot assign parent object to child variable.

7. What will be the output of the following Java code?

public class Shape

{

public int area()

{

return 1;

}

}

public class Square extends Shape

{

public int area()

{

return 2;

}

}

public class Rectangle extends Shape

{

public int area()

{

return 3;

}

}

class Main()

{

public static void main(String[] args)

{

Shape shape = new Shape();

Square square = new Square();

Rectangle rect = new Rectangle();

rect = (Rectangle)square;

System.out.println(square.area());

}

}

a) Compilation failure

b) 3

c) Runtime Exception

d) 2

Answer: a

Explanation: We cannot assign one child class object to another child class variable.

interface Shape

{

public int area();

}

public class Square implements Shape

{

public int area()

{

return 2;

}

}

public class Rectangle implements Shape

{

public int area()

{

return 3;

}

}

8. What will be the output of the following Java code?

public class Shape

{

public int area()

{

return 1;

}

}

public class Square extends Shape

{

public int area()

{

return 2;

}

}

public class Rectangle extends Shape

{

public int area()

{

return 3;

}

}

class Main()

{

public static void main(String[] args)

{

Shape shape = new Shape();

Square square = new Square();

Rectangle rect = new Rectangle();

rect = (Rectangle)square;

System.out.println(square.area());

}

}

a) Compilation failure

b) 3

c) Runtime Exception

d) 2

Answer: a

Explanation: Interface cannot be instantiated. So we cannot create instances of shape.

9. What will be the output of the following Java code?

public class Shape

{

public int area()

{

return 1;

}

}

public class Square extends Shape

{

public int area()

{

return 2;

}

}

public class Rectangle extends Shape

{

public int area()

{

return 3;

}

}

public static void main(String[] args)

{

Shape shape = new Square();

shape = new Rectangle();

System.out.println(shape.area());

}

a) Compilation failure

b) 3

c) Runtime Exception

d) 2

Answer: b

Explanation: With parent class variable we can access methods declared in parent class. If the parent class variable is assigned child class object than it accesses the method of child class.

10. What will be the output of the following Java code?

public class Shape

{

public int area()

{

return 1;

}

}

public class Square extends Shape

{

public int area()

{

return 2;

}

}

public class Rectangle extends Shape

{

public int area()

{

return 3;

}

}

public static void main(String[] args)

{

Shape square = new Square();

Shape rect = new Rectangle();

square = rect;

System.out.println(square.area());

}

a) Compilation failure

b) 3

c) Runtime Exception

d) 2

Answer: b

Explanation: The method of the child class object is accessed. When we reassign objects, the methods of the latest assigned object are accessed.

6 – Coding best practices

1. What should the return type of method where there is no return value?

a) Null

b) Empty collection

c) Singleton collection

d) Empty String

Answer: b

Explanation: Returning Empty collection is a good practice. It eliminates chances of unhandled null pointer exceptions.

2. What data structure should be used when number of elements is fixed?

a) Array

b) Array list

c) Vector

d) Set

Answer: a

Explanation: Array list has variable size. Array is stored in contiguous memory. Hence, reading is faster. Also, array is memory efficient.

3. What causes the program to exit abruptly and hence its usage should be minimalistic?

a) Try

b) Finally

c) Exit

d) Catch

Answer: c

Explanation: In case of exit, the program exits abruptly hence would never be able to debug the root cause of the issue.

4. Which of the following is good coding practice to determine oddity?

i)

public boolen abc(int num)

{

return num % 2 == 1;

}

ii)

public boolean xyz(int num)

{

return (num & 1)!= 0;

}

a) i

b) ii

c) (i) causes compilation error

d) (ii) causes compilation error

Answer: b

Explanation: Arithmetic and logical operations are much faster than division and multiplication.

5. Which one of the following causes memory leak?

a) Release database connection when querying is complete

b) Use Finally block as much as possible

c) Release instances stored in static tables

d) Not using Finally block often

Answer: d

Explanation: Finally block is called in successful as well exception scenarios. Hence, all the connections are closed properly which avoids memory leak.

6. Which of the following is a best practice to measure time taken by a process for execution?

a) System.currentTimeMillis()

b) System.nanoTime()

c) System.getCurrentTime()

d) System.getProcessingTime()

Answer: b

Explanation: System.nanoTime takes around 1/100000 th of a second whereas System.currentTimeMillis takes around 1/1000th of a second.

7. What one of the following is best practice to handle Null Pointer exception?

i) int noOfStudents = line.listStudents().count;

ii) int noOfStudents = getCountOfStudents(line);

public int getCountOfStudents(List line)

{

if(line != null)

{

if(line.listOfStudents() != null)

{

return line.listOfStudents().size();

}

}

throw new NullPointerException("List is empty");

}

a) Option (i)

b) Option (ii)

c) Compilation Error

d) Option (ii) gives incorrect result

Answer: b

Explanation: Null check must be done while dealing with nested structures to avoid null pointer exceptions.

8. Which of the below is true about java class structure?

a) The class name should start with lowercase

b) The class should have thousands of lines of code

c) The class should only contain those attribute and functionality which it should; hence keeping it short

d) The class attributes and methods should be public

Answer: c

Explanation: Class name should always start with upper case and contain those attribute and functionality which it should (Single Responsibility Principle); hence keeping it short. The attributes should be usually private with get and set methods.

9. Which of the below is false about java coding?

a) variable names should be short

b) variable names should be such that they avoid ambiguity

c) test case method names should be created as english sentences without spaces

d) class constants should be used when we want to share data between class methods

Answer: a

Explanation: variable names like i, a, abc, etc should be avoided. They should be real world names which avoid ambiguity. Test case name should explain its significance.

10. Which is better in terms of performance for iterating an array?

a) for(int i=0; i<100; i++)

b) for(int i=99; i>=0; i–)

c) for(int i=100; i<0; i++)

d) for(int i=99; i>0; i++)

Answer: b

Explanation: reverse traversal of array take half number cycles as compared to forward traversal. The other for loops will go in infinite loop.