Here’s a Multiple-Choice Question (MCQ) bank based on the notes:

Reflection in Java

MCQs

1. What is the primary use of Java Reflection?
    - A. To modify JVM runtime behavior
    - B. To examine and manipulate classes, fields, and methods at runtime
    - C. To improve performance during compilation
    - D. To create anonymous classes dynamically
    - *Answer*: B
2. Which package contains the classes for reflection?
    - A. java.util.reflect
    - B. java.lang
    - C. java.lang.reflect
    - D. java.reflect
    - *Answer*: C
3. Which of these is NOT a reflection class in `java.lang.reflect`?
    - A. Method
    - B. Field
    - C. Constructor
    - D. Wrapper
    - *Answer*: D
4. The method `setAccessible(true)` is used to:
    - A. Override a method at runtime
    - B. Allow access to private or protected members
    - C. Make a method synchronized
    - D. Enable reflection on final fields
    - *Answer*: B
5. Which is a disadvantage of using reflection in Java?
    - A. It cannot handle annotations
    - B. It breaks platform independence
    - C. It introduces performance overhead
    - D. It lacks runtime type checking
    - *Answer*: C

Static Keyword in Java

MCQs

6. Which of the following is true about static members in Java?
    - A. Static members are allocated memory for every object.
    - B. Static members are tied to the instance of the class.
    - C. Static members are shared across all instances of the class.
    - D. Static members cannot be accessed without an instance.

- *Answer*: C

7. Which of these cannot be declared as static?
    - A. Methods
    - B. Variables
    - C. Blocks
    - D. Classes
    - *Answer*: D
8. Static methods in Java:
    - A. Can use the `this` keyword
    - B. Can access static members directly
    - C. Must be private
    - D. Can only be overridden
    - *Answer*: B
9. The static block in Java:
    - A. Executes every time an object is created
    - B. Executes when the class is loaded
    - C. Executes after the main method
    - D. Cannot modify static variables
    - *Answer*: B
10. Which of these is NOT allowed in a static method?
    - A. Accessing static fields
    - B. Using `this` or `super`
    - C. Invoking static methods
    - D. Returning static variables
    - *Answer*: B

Final Keyword in Java

MCQs

11. A final variable in Java:
    - A. Can only be a static variable
    - B. Can be reassigned once initialized
    - C. Cannot be reassigned after initialization
    - D. Must always be initialized during declaration
    - *Answer*: C
12. Which of the following statements is correct for a final method?
    - A. It cannot be overloaded.
    - B. It can be overridden by a subclass.
    - C. It cannot be overridden by a subclass.
    - D. It must be declared abstract.
    - *Answer*: C
13. What happens when a class is declared as final?
    - A. It cannot have subclasses.

- B. It cannot have static members.
- C. It cannot contain constructors.
- D. It can only contain final variables.
- *Answer*: A

14. A blank final variable:

- A. Must be initialized during declaration.
- B. Can be initialized in the constructor.
- C. Can be modified in a static block.
- D. Cannot be declared as static.
- *Answer*: B

15. Which is NOT true about final reference variables?

- A. They cannot refer to a different object.
- B. The object they reference cannot be modified.
- C. They must be initialized at the time of declaration.
- D. They refer to memory location rather than object mutability.
- *Answer*: B

Super and This Keywords in Java

MCQs

16. The `super` keyword in Java:

- A. Refers to the current class object
- B. Refers to the immediate superclass object
- C. Refers to a static method in the same class
- D. Refers to the final parent class object
- *Answer*: B

17. Which of the following is a valid use of `super` in Java?

- A. Accessing static methods of the superclass
- B. Accessing private members of the superclass
- C. Accessing overridden methods of the superclass
- D. Referring to the current class object
- *Answer*: C

18. The `this` keyword:

- A. Refers to the immediate parent class object
- B. Refers to the current class object
- C. Refers to a static block in the current class
- D. Refers to a static method in the current class
- *Answer*: B

19. Which of these is NOT true about `super` and `this` keywords?

- A. `this` refers to the current class object, while `super` refers to the superclass object.
- B. Both can be used in static methods.
- C. Both are used in constructor chaining.

- D. `this` calls constructors in the same class, while `super` calls constructors in the superclass.
- *Answer*: B

20. Which statement is incorrect about the use of `this` in Java?

- A. It can refer to instance variables of the current class.
- B. It can be used to call non-static methods of the current class.
- C. It can be used in static blocks.
- D. It can be passed as an argument to a method.
- *Answer*: C

Here's a **Output-Based MCQ Question Bank** derived from the notes:

Reflection in Java

MCQs

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

```
import java.lang.reflect.Method;

public class TestReflection {
    public static void main(String[] args) throws Exception {
        Class<?> c = Class.forName("java.lang.String");
        Method[] methods = c.getDeclaredMethods();
        System.out.println(methods[0].getName());
    }
}
```

- A. The name of the first method declared in the `String` class
- B. `null`
- C. Throws a `ClassNotFoundException`
- D. Prints "String"
- *Answer*: A

2. What happens if the following code tries to access a private field using reflection without setting `setAccessible(true)`?

```
import java.lang.reflect.Field;

public class TestReflection {
    private int value = 10;

    public static void main(String[] args) throws Exception {
        TestReflection obj = new TestReflection();
        Field field = obj.getClass().getDeclaredField("value");
        System.out.println(field.get(obj));
    }
}
```

- A. Outputs `10`
- B. Throws `IllegalAccessException`

- C. Outputs `null`
- D. Throws `NoSuchFieldException`
- *Answer*: B

Static Keyword in Java

MCQs

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

```
public class TestStatic {
    static int x = 10;
    public static void main(String[] args) {
        x = 20;
        System.out.println(x);
    }
}
```

- A. 10
- B. 20
- C. Compilation Error
- D. Runtime Exception
- *Answer*: B

4. What will happen if the following code is executed?

```
public class TestStaticBlock {
    static {
        System.out.println("Static block executed");
    }
    public static void main(String[] args) {
        System.out.println("Main method executed");
    }
}
```

- A. Only "Main method executed" is printed.
- B. Only "Static block executed" is printed.
- C. "Static block executed" is printed first, followed by "Main method executed".
- D. Compilation Error
- *Answer*: C

5. What is the output of the following code?

```
public class StaticExample {
    static int a;
    public static void main(String[] args) {
        System.out.println(a);
    }
}
```

- A. `0`
- B. `null`

- C. Compilation Error
- D. Uninitialized variable exception
- *Answer**: A

Final Keyword in Java

MCQs

6. What is the output of the following code?

```
public class FinalExample {
    final int x = 10;
    public static void main(String[] args) {
        FinalExample obj = new FinalExample();
        System.out.println(obj.x);
    }
}
```

- A. 0
- B. 10
- C. Compilation Error
- D. Runtime Exception
- *Answer**: B

7. What will happen in this case?

```
public class FinalVariable {
    final int a = 5;
    public void changeValue() {
        a = 10;
    }
}
```

- A. Outputs `10`
- B. Compilation Error
- C. Runtime Exception
- D. Throws a `NullPointerException`
- *Answer**: B

8. What is the output of the following code snippet?

```
public class BlankFinalVariable {
    final int x;
    public BlankFinalVariable() {
        x = 42;
    }
    public static void main(String[] args) {
        BlankFinalVariable obj = new BlankFinalVariable();
        System.out.println(obj.x);
    }
}
```

- A. Compilation Error

- B. 0
- C. 42
- D. Runtime Exception
- *Answer**: C

Super and This Keywords

MCQs

9. What is the output of the following code?

```
class Parent {
    int x = 10;
}
class Child extends Parent {
    int x = 20;
    void display() {
        System.out.println(super.x);
        System.out.println(this.x);
    }
}
public class TestSuper {
    public static void main(String[] args) {
        Child obj = new Child();
        obj.display();
    }
}
```

- A. 20 10
- B. 10 10
- C. 10 20
- D. 20 20
- *Answer**: C

10. What will happen in the following case?

```
class Parent {
    Parent() {
        System.out.println("Parent Constructor");
    }
}
class Child extends Parent {
    Child() {
        super();
        System.out.println("Child Constructor");
    }
}
public class TestConstructor {
    public static void main(String[] args) {
        new Child();
    }
}
```

- A. Only "Child Constructor" is printed.
- B. Only "Parent Constructor" is printed.
- C. "Parent Constructor" is printed first, followed by "Child Constructor".
- D. Compilation Error
- *Answer**: C

11. What will be the output of this program?

```
class A {
    void method() {
        System.out.println("Parent method");
    }
}
class B extends A {
    void method() {
        super.method();
        System.out.println("Child method");
    }
}
public class TestOverride {
    public static void main(String[] args) {
        B obj = new B();
        obj.method();
    }
}
```

- A. "Parent method"
- B. "Child method"
- C. "Parent method" followed by "Child method"
- D. Compilation Error
- *Answer**: C

Here is a multiple-choice question (MCQ) bank based on the content of the document:

MCQs on Wrapper Classes in Java

Basics of Wrapper Classes

1. What is a wrapper class in Java?
   a) A class that wraps an object into a primitive value.
   b) A class that wraps a primitive data type into an object.
   c) A class used to convert strings to numbers.
   d) A class used for only object-oriented operations.
   Answer: b

2. Which of the following is NOT a wrapper class?
   a) Integer
   b) Double
   c) String
   d) Character
   Answer: c

3. Why are wrapper classes used in Java?
   a) To improve the performance of primitive types.
   b) To allow primitive types to be used where objects are required.
   c) To create custom data structures.
   d) To provide faster arithmetic operations.
   Answer: b

Wrapping and Unwrapping

4. How do you wrap an int value into an `Integer` object?
   a) `Integer obj = Integer.wrap(x);`
   b) `Integer obj = new Integer(x);`
   c) `Integer obj = Integer.valueOf(x);`
   d) `Integer obj = x;`
   Answer: b

5. What method is used to unwrap an `Integer` object into a primitive int?
   a) `getValue()`
   b) `unwrap()`
   c) `intValue()`
   d) `primitive()`
   Answer: c

6. What is the output of the following code?

```
Integer obj = 50;
int num = obj;
System.out.println(num);
```

   a) 50
   b) 0
   c) Error at compile time
   d) NullPointerException
   Answer: a

Autoboxing and Unboxing

7. What is autoboxing in Java?
   a) Automatic conversion of wrapper objects to primitive types.
   b) Automatic conversion of primitive types to wrapper objects.
   c) Wrapping primitive values manually.
   d) Converting wrapper objects to strings.
   Answer: b

8. What is unboxing in Java?
   a) Extracting primitive values from wrapper objects.
   b) Wrapping primitive values into objects.
   c) Converting objects into strings.
   d) A feature for working with arrays.
   Answer: a

9. Which version of Java introduced autoboxing and unboxing?
   a) Java 1.4

b) Java 5.0
c) Java 7.0
d) Java 8.0
Answer: b

List of Wrapper Classes

10. What is the wrapper class for the `float` primitive type?
    a) Double
    b) Float
    c) Decimal
    d) Long
    Answer: b

11. Which of the following is true about wrapper classes?
    a) They are mutable.
    b) They are defined in the `java.util` package.
    c) They are immutable and final.
    d) They can override constructors.
    Answer: c

12. Which wrapper class does not have a constructor that accepts a `String` as an argument?
    a) Integer
    b) Boolean
    c) Character
    d) Double
    Answer: c

Key Features

13. Which of the following is NOT a feature of wrapper classes?
    a) Provide methods for data manipulation.
    b) Allow overriding of methods.
    c) Useful for converting primitives to objects.
    d) Objects of wrapper classes are immutable.
    Answer: b

14. Which package contains all the wrapper classes in Java?
    a) java.util
    b) java.io
    c) java.lang
    d) java.base
    Answer: c

15. What happens when you attempt to modify the internal value of a wrapper class object?
    a) The value is updated.
    b) A new object is created with the updated value.
    c) A runtime error occurs.
    d) The operation is ignored.
    Answer: b

Advanced Questions

16. Which of the following statements is true about autoboxing?
    a) It is slower than manually wrapping primitive values.
    b) It occurs implicitly at compile time.
    c) It is the opposite of unboxing.
    d) It cannot be used with arrays.
    Answer: b

17. What is the main limitation of wrapper classes?
    a) Increased memory usage compared to primitive types.
    b) Inability to use them in collections.
    c) Lack of type safety.
    d) Absence of utility methods.
    Answer: a

18. Which wrapper class is used to represent the `boolean` primitive type?
    a) Boolean
    b) Bool
    c) Byte
    d) Bit
    Answer: a

Here is a set of **output-based questions** for Java wrapper classes. Each question includes a code snippet and possible outputs to test understanding of concepts such as autoboxing, unboxing, and wrapping/unwrapping.

Output-Based Questions

Basic Wrapping and Unwrapping

1. **Code**:

```
public class Test {
    public static void main(String[] args) {
        int x = 10;
        Integer obj = new Integer(x);
        System.out.println(obj);
    }
}
```

    Options: a) 10
    b) `Integer@hashcode`
    c) Compilation error
    d) NullPointerException
    Answer: a

2. **Code**:

```
public class Test {
    public static void main(String[] args) {
        Integer obj = new Integer(100);
        int value = obj.intValue();
        System.out.println(value + 50);
```

```
    }
}
```

Options: a) 100
b) 150
c) 50
d) Compilation error
Answer: b

Autoboxing and Unboxing

3. **Code**:

```
public class Test {
    public static void main(String[] args) {
        Integer obj = 200; // Autoboxing
        int value = obj;   // Unboxing
        System.out.println(value + obj);
    }
}
```

Options: a) 200
b) 400
c) Compilation error
d) NullPointerException
Answer: b

4. **Code**:

```
public class Test {
    public static void main(String[] args) {
        Integer obj = null;
        int value = obj; // Unboxing null
        System.out.println(value);
    }
}
```

Options: a) 0
b) NullPointerException
c) Compilation error
d) Undefined behavior
Answer: b

Immutable Wrapper Classes

5. **Code**:

```
public class Test {
    public static void main(String[] args) {
        Integer obj1 = 50;
        Integer obj2 = obj1;
        obj1 = obj1 + 10;
```

```
        System.out.println(obj1 + " " + obj2);
    }
}
```

Options: a) 60 60
b) 60 50
c) Compilation error
d) Undefined behavior
Answer: b

6. **Code**:

```
public class Test {
    public static void main(String[] args) {
        Boolean obj = true;
        if (obj) {
            System.out.println("Wrapper works!");
        } else {
            System.out.println("Wrapper failed!");
        }
    }
}
```

Options: a) Wrapper works! b) Wrapper failed! c) Compilation error
d) NullPointerException
Answer: a

Parsing and Conversion

7. **Code**:

```
public class Test {
    public static void main(String[] args) {
        String str = "123";
        int num = Integer.parseInt(str);
        System.out.println(num + 10);
    }
}
```

Options: a) 123
b) 133
c) Compilation error
d) NumberFormatException
Answer: b

8. **Code**:

```
public class Test {
    public static void main(String[] args) {
        String str = "abc";
        int num = Integer.parseInt(str);
        System.out.println(num);
```

```
    }
}
```

Options: a) 0
b) Compilation error
c) NumberFormatException
d) Undefined behavior
Answer: c

Comparisons and Equality

9. **Code**:

```
public class Test {
    public static void main(String[] args) {
        Integer obj1 = 100;
        Integer obj2 = 100;
        System.out.println(obj1 == obj2);
    }
}
```

Options: a) true
b) false
c) Compilation error
d) NullPointerException
Answer: a

10. **Code**:

```
public class Test {
    public static void main(String[] args) {
        Integer obj1 = 1000;
        Integer obj2 = 1000;
        System.out.println(obj1 == obj2);
    }
}
```

Options: a) true
b) false
c) Compilation error
d) NullPointerException
Answer: b
(Explanation: Objects are compared outside the integer cache range of -128 to 127.)

Mixing Types

11. **Code**:

```
public class Test {
    public static void main(String[] args) {
        Double obj = 10.5;
```

```
        System.out.println(obj.intValue());
    }
}
```

Options: a) 10.5
b) 10
c) Compilation error
d) 11
Answer: b

12. **Code**:

```
public class Test {
    public static void main(String[] args) {
        Float obj = 3.14f;
        System.out.println(obj instanceof Object);
    }
}
```

Options: a) true
b) false
c) Compilation error
d) NullPointerException
Answer: a

Advanced Use Cases

13. **Code**:

```
public class Test {
    public static void main(String[] args) {
        Integer obj = Integer.valueOf("256");
        System.out.println(obj + Integer.valueOf("10"));
    }
}
```

Options: a) 25610
b) 266
c) Compilation error
d) NumberFormatException
Answer: b

14. **Code**:

```
public class Test {
    public static void main(String[] args) {
        Boolean obj1 = new Boolean("true");
        Boolean obj2 = new Boolean("false");
        System.out.println(obj1 && obj2);
    }
}
```

Options: a) true
b) false
c) Compilation error
d) Undefined behavior
Answer: b

Topic: ENUM in Java

1. Which of the following is true about enum in Java?
   a) An enum can extend a class
   b) An enum is implicitly final
   c) An enum can implement interfaces
   d) Both b and c
   Answer: d

2. What does the `values()` method of an enum return?
   a) The name of the enum constant
   b) An array of all enum constants
   c) The ordinal value of the enum constant
   d) None of the above
   Answer: b

3. Which of the following is invalid for enums?
   a) Declaring constructors
   b) Implementing interfaces
   c) Using inheritance
   d) Declaring methods
   Answer: c

Topic: Exception Handling in Java

4. What is the purpose of the `throws` keyword in Java?
   a) To throw multiple exceptions explicitly
   b) To declare exceptions that a method might throw
   c) To handle runtime errors
   d) None of the above
   Answer: b

5. Which of the following blocks in exception handling is always executed?
   a) `try`
   b) `catch`
   c) `finally`
   d) None of the above
   Answer: c

6. Checked exceptions in Java are:
   a) Detected at runtime
   b) Detected at compile-time and handled at runtime
   c) Handled automatically by the JVM
   d) None of the above
   Answer: b

Topic: File Handling in Java

7. Which of the following streams is used to read character data from a file?
   a) `FileReader`
   b) `BufferedReader`
   c) Both a and b
   d) None of the above
   Answer: c

8. What does the `createNewFile()` method of the `File` class return if the file already exists?
   a) `true`
   b) `false`
   c) Throws an exception
   d) None of the above
   Answer: b

9. Which class is used for line-by-line reading of a text file in Java?
   a) `FileReader`
   b) `BufferedReader`
   c) `PrintWriter`
   d) None of the above
   Answer: b

Topic: Garbage Collection in Java

10. Which method is used to request garbage collection in Java?
    a) `System.collect()`
    b) `Runtime.collect()`
    c) `System.gc()`
    d) None of the above
    Answer: c

11. What is the purpose of the `finalize()` method?
    a) To clean up resources before an object is garbage collected
    b) To reclaim unused memory
    c) To handle `OutOfMemoryError`
    d) None of the above
    Answer: a

12. Which of the following is true regarding garbage collection?
    a) Programmers can control when GC is executed
    b) GC guarantees immediate memory cleanup
    c) GC is triggered when JVM runs low on memory
    d) None of the above
    Answer: c

Topic: Generics in Java

13. What is the main advantage of using generics in Java?
    a) Runtime safety
    b) Type safety at compile-time

c) Faster execution
d) Reduces memory usage
Answer: b

14. Which of the following is a valid generic class declaration?
    a) `class MyClass<T>`
    b) `class MyClass(int T)`
    c) `class MyClass<?>`
    d) `class MyClass<extends T>`
    Answer: a

15. Which wildcard can accept only objects of a specific class and its subclasses?
    a) `<?>`
    b) `<? extends X>`
    c) `<? super X>`
    d) None of the above
    Answer: b

Topic: Inner Classes in Java

16. Which of the following is true about an inner class?
    a) An inner class can have static methods
    b) An inner class can access private members of its outer class
    c) An inner class must be static
    d) None of the above
    Answer: b

17. What is an anonymous inner class?
    a) A static nested class
    b) A class declared without a name
    c) A top-level class
    d) None of the above
    Answer: b

18. Which of the following types of inner classes is considered a top-level nested class?
    a) Static nested class
    b) Anonymous inner class
    c) Method-local inner class
    d) Regular inner class
    Answer: a

Here are **output-based multiple-choice questions** derived from the notes and examples:

Topic: ENUM in Java

1. What is the output of the following code?

```
enum Day { MON, TUE, WED; }
public class Test {
    public static void main(String[] args) {
        for (Day d : Day.values()) {
```

```
            System.out.print(d + " ");
        }
    }
}
```

a) MON TUE WED
b) MON, TUE, WED
c) Compilation error
d) None of the above
Answer: a

2. What happens if you call `toString()` on an enum constant?

```
enum Color { RED, BLUE, GREEN; }
public class Test {
    public static void main(String[] args) {
        System.out.println(Color.RED.toString());
    }
}
```

a) Prints `RED`
b) Prints the ordinal value of `RED`
c) Compilation error
d) Throws `NullPointerException`
Answer: a

Topic: Exception Handling in Java

3. What is the output of the following code?

```
public class Test {
    public static void main(String[] args) {
        try {
            int result = 10 / 0;
        } catch (ArithmeticException e) {
            System.out.print("Exception Caught");
        } finally {
            System.out.print(" Finally Block");
        }
    }
}
```

a) Exception Caught
b) Exception Caught Finally Block
c) Compilation error
d) None of the above
Answer: b

4. What is the output of the following program?

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

```
            int[] arr = new int[5];
            System.out.println(arr[10]);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.print("Index Issue ");
        } finally {
            System.out.print("Clean Up");
        }
    }
}
```

a) Index Issue
b) Clean Up
c) Index Issue Clean Up
d) Runtime Exception
Answer: c

Topic: File Handling in Java

5. What is the output of the following code?

```
import java.io.*;
public class Test {
    public static void main(String[] args) throws IOException {
        File f = new File("demo.txt");
        System.out.print(f.exists());
        f.createNewFile();
        System.out.print(f.exists());
    }
}
```

Assuming `demo.txt` does not exist initially:
a) false true
b) true true
c) false false
d) Compilation error
Answer: a

6. What will happen if the following program is executed?

```
import java.io.*;
public class Test {
    public static void main(String[] args) throws IOException {
        FileReader fr = new FileReader("nonexistent.txt");
        System.out.println("File Opened");
    }
}
```

a) Prints "File Opened"
b) Compilation error
c) Throws `FileNotFoundException`
d) None of the above
Answer: c

Topic: Garbage Collection in Java

7. What is the output of the following code?

```java
public class Test {
    public static void main(String[] args) {
        Test t = new Test();
        t = null;
        System.gc();
        System.out.print("GC Called");
    }
    @Override
    protected void finalize() {
        System.out.print(" Finalized");
    }
}
```

a) GC Called Finalized
b) Finalized GC Called
c) GC Called
d) Finalized
Answer: c (Finalization depends on JVM behavior)

8. What happens if `System.gc()` is called explicitly?

```java
public class Test {
    public static void main(String[] args) {
        System.gc();
        System.out.println("End of Program");
    }
}
```

a) Garbage collector is guaranteed to run
b) JVM decides whether to run GC
c) Throws `RuntimeException`
d) None of the above
Answer: b

Topic: Generics in Java

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

```java
import java.util.*;
public class Test {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("A");
        list.add("B");
        list.add("C");
        for (String s : list) {
            System.out.print(s + " ");
        }
```

```
    }
}
```

a) A B C
b) Compilation error
c) Runtime exception
d) None of the above
Answer: a

10. What happens when the following code is executed?

```
import java.util.*;
public class Test {
    public static void main(String[] args) {
        ArrayList<?> list = new ArrayList<>();
        list.add("A");
        System.out.print(list.size());
    }
}
```

a) 1
b) 0
c) Compilation error
d) Runtime exception
Answer: c (Cannot add elements to a wildcard list)

Topic: Inner Classes in Java

11. What is the output of the following program?

```
public class Outer {
    class Inner {
        void display() {
            System.out.print("Hello Inner");
        }
    }
    public static void main(String[] args) {
        Outer o = new Outer();
        Outer.Inner i = o.new Inner();
        i.display();
    }
}
```

a) Hello Inner
b) Compilation error
c) Runtime exception
d) None of the above
Answer: a

12. What will happen if the following code is executed?

```
public class Outer {
    static class Inner {
```

```
        void display() {
            System.out.print("Hello Static Nested");
        }
    }
    public static void main(String[] args) {
        Outer.Inner i = new Outer.Inner();
        i.display();
    }
}
```

a) Hello Static Nested
b) Compilation error
c) Runtime exception
d) None of the above
Answer: a

Here is a multiple-choice question bank for the topics covered in the notes:

Topic: JDBC in Java

Basic Concepts

1. What does JDBC stand for?
    - A) Java Database Connection
    - B) Java Database Communication
    - C) Java Database Connectivity
    - D) Java Driver Communication
    - *Answer**: C
2. Which component of JDBC translates Java method calls into database-specific calls?
    - A) JDBC Client
    - B) JDBC API
    - C) JDBC Driver
    - D) JDBC Manager
    - *Answer**: C
3. Which type of JDBC driver is also known as the "thin driver"?
    - A) Type 1
    - B) Type 2
    - C) Type 3
    - D) Type 4
    - *Answer**: D

Architecture

4. The JDBC API layer connects which two components?
    - A) Application and JDBC Manager
    - B) JDBC Manager and Database
    - C) Application and Database

- D) Application and Network Server
- *Answer*: A

5. What does the `ResultSet.next()` method do?

   - A) Moves the cursor to the next record
   - B) Deletes the current record
   - C) Moves the cursor to the first record
   - D) Checks if the ResultSet is empty
   - *Answer*: A

Topic: Method Overloading and Overriding

Overloading

6. Which of the following is NOT valid for method overloading?

   - A) Different parameter types
   - B) Different number of parameters
   - C) Changing the return type only
   - D) Changing the parameter sequence
   - *Answer*: C

7. What is method overloading also known as?

   - A) Compile-time polymorphism
   - B) Runtime polymorphism
   - C) Late binding
   - D) Dynamic dispatch
   - *Answer*: A

Overriding

8. Which access modifier allows overriding?

   - A) private
   - B) final
   - C) protected
   - D) static
   - *Answer*: C

9. What happens if the `@Override` annotation is not used?

   - A) The compiler checks overriding rules
   - B) The method cannot be overridden
   - C) Only three basic rules are checked
   - D) The superclass method is deleted
   - *Answer*: C

10. Method overriding supports which type of polymorphism?

    - A) Compile-time
    - B) Runtime
    - C) Static
    - D) Early binding
    - *Answer*: B

Topic: Packages in Java

General Concepts

11. What is a package in Java?

    - A) A collection of variables
    - B) A physical folder structure containing related classes/interfaces
    - C) A tool for debugging Java code
    - D) A temporary storage for objects
    - *Answer*: B

12. Which of the following is NOT an advantage of using packages?

    - A) Resolves naming conflicts
    - B) Provides access protection
    - C) Allocates memory to imported classes
    - D) Improves maintainability
    - *Answer*: C

Types of Packages

13. Which of these is an example of a user-defined package?

    - A) `java.util`
    - B) `java.lang`
    - C) `myPackage`
    - D) `javax.swing`
    - *Answer*: C

14. Which is true about predefined packages in Java?

    - A) Start with `java` or `javax`
    - B) Always require explicit installation
    - C) Cannot be used in custom applications
    - D) Are defined by third-party developers only
    - *Answer*: A

Key Points

15. How many package declarations can a Java class have?

    - A) Unlimited
    - B) One
    - C) Two
    - D) None
    - *Answer*: B

16. What is the syntax for importing a specific class from a package?

    - A) `import packageName.*;`
    - B) `include packageName.className;`
    - C) `import packageName.className;`
    - D) `load packageName.className;`
    - *Answer*: C

Topic: Type Casting in Java

17. Which of the following is an example of implicit type casting?

    - A) `int x = 10; double y = x;`
    - B) `double x = 5.6; int y = (int)x;`
    - C) `float x = 3.14; int y = x;`
    - D) `char c = 65; int y = (int)c;`
    - *Answer*: A

18. What happens during explicit type casting?

    - A) Automatic type conversion
    - B) Programmer-defined type conversion
    - C) Loss of data cannot occur
    - D) No syntax is required
    - *Answer*: B

19. What is the process of converting a subclass type to a superclass type?

    - A) Downcasting
    - B) Generalization (Upcasting)
    - C) Narrowing
    - D) Specialization
    - *Answer*: B
