



PROGRAMMING IN JAVA

Assignment 01

TYPE OF QUESTION: MCQ

Number of questions: 10

Total marks: $10 \times 1 = 10$

QUESTION 1:

Which of the following is true?

- a. Java uses only interpreter.
- b. Java uses only compiler.
- c. Java uses both interpreter and compiler.
- d. None of the above.

Correct Answer:

- c. Java uses both interpreter and compiler.

Detailed Solution:

Creating a .class file from .java using javac command is a compilation task, whereas execution of a .class file using java is the process of interpretation.



QUESTION 2:

A Java file with extension '.class' contains

- a. Java source code
- b. HTML tags
- c. Java Byte code
- d. A program file written in Java programming language

Correct Answer:

- c. Java Byte code

Detailed Solution:

A .class file is a complied version of the .java file in byte code (it is a kind of object code with JVM (Java Virtual Machine) as the target machine.)



QUESTION 3:

Which of the following is not an object-oriented programming paradigm?

- a. Encapsulation
- b. Inheritance
- c. Polymorphism
- d. Dynamic memory allocation

Correct Answer:

- d. Dynamic memory allocation

Detailed Solution:

Dynamic memory allocation is a memory allocation strategy and not a programming paradigm.



QUESTION 4:

Java is a platform independent programming language because

- a. It compiles an intermediate code targeting a virtual machine, which can be interpreted by an interpreter for a given OS.
- b. The Java compiler translates the source code directly into the machine-level language.
- c. It follows the concept of “write once and compile everywhere”.
- d. It is written almost similar to the English language.

Correct Answer:

- a. It compiles to an intermediate code targeting a virtual machine, which can be interpreted by an interpreter for a given OS.

Detailed Solution:

The compiled code (byte code) can be executed (interpreted) on any platform running a JVM.



QUESTION 5:

Which of the following is not a Language Processor?

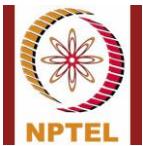
- a. Assembler
- b. Compiler
- c. Interpreter
- d. Fortran

Correct Answer:

- d. Fortran

Detailed Solution:

A computer understands instructions in machine code i.e., in the form of 0s and 1s. Special translators are required for this operation like Assembler, Compiler and Interpreter. Fortran is a programming language but not a language processor.



QUESTION 6:

A platform is the hardware or software environment in which a program runs. Which of the following is/are Java platform component(s)?

- a. HTML
- b. Java Virtual Machine
- c. Javascript
- d. HotJava

Correct Answer:

- b. Java Virtual Machine

Detailed Solution:

"A platform is the hardware or software environment in which a program runs. Some of the most popular platforms are Microsoft Windows, Linux, Solaris OS, and Mac OS. Most platforms can be described as a combination of the operating system and underlying hardware. The Java platform differs from other platforms as it is a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

The Java Virtual Machine

The Java Application Programming Interface (API)



QUESTION 7:

What is the correct sequence of steps to execute a Java program?

- I. **Compile the Program:** Use the `javac` command to compile the code into bytecode.
- II. **Edit the Program:** Write the code in a text editor or IDE.
- III. **Run the Program:** Use the `java` command to execute the bytecode.

Which of the following options represents this sequence?

- a. Run → Edit → Compile
- b. Edit → Run → Compile
- c. Compile → Edit → Run
- d. Edit → Compile → Run

Correct Answer:

- d. Edit → Compile → Run

Detailed Solution:

The Java development process involves writing code (Edit), converting it to bytecode (Compile), and then executing it on the JVM (Run).



QUESTION 8:

What is the primary difference between javac and java commands?

- a. **javac is used to edit Java code, while java runs Java programs.**
- b. **javac compiles Java source code to bytecode, while java executes the bytecode on the JVM.**
- c. **javac executes Java programs, while java is used for compilation.**
- d. **Both are used for compiling Java programs.**

Correct Answer:

- b. **javac compiles Java source code to bytecode, while java executes the bytecode on the JVM.**

Detailed Solution:

The javac command converts .java source files into .class bytecode files.

The java command executes the .class file using the Java Virtual Machine (JVM).



QUESTION 9:

Which of the following is not a feature of Java?

- a. Platform Independence
- b. Object-Oriented Programming
- c. Supports Explicit Pointers
- d. Supports Polymorphism

Correct Answer:

- c. Supports Explicit Pointers

Detailed Solution:

Java is platform-independent, object-oriented, and secure. It does not support explicit pointers, as they can lead to memory management issues and compromise security.



QUESTION 10:

What is the output of the following code?

```
class NPTEL {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

- a. Hello, World!
- b. HelloWorld!
- c. Compilation Error
- d. Runtime Error

Correct Answer:

- a. Hello, World!

Detailed Solution:

Java program to print Hello, World!



PROGRAMMING IN JAVA

Assignment 02

TYPE OF QUESTION: MCQ

Number of questions: 10

Total marks: $10 \times 1 = 10$

QUESTION 1:

Consider the following object declaration statement

```
Scanner in = new Scanner(System.in);
```

What does `System.in` stands for in the above declaration?

- a. Any file storing data
- b. Refers to the *standard input stream*, which is typically the keyboard by default.
- c. Reference to scanner as an input device
- d. It is a mouse as an input device

Correct Answer:

- b. Refers to the *standard input stream*, which is typically the keyboard by default.

Detailed Solution:

`System.in` refers to the standard input device and the keyboard is treated as standard input device. Thus, the code implies reading a data from keyboard.

Please refer to chapter 3 of book Joy With Java for a more detailed explanation.



QUESTION 2:

What will be the output of the following Java program?

```
public class VarPrint {  
    int x = 30;  
    static int y = 20;  
  
    public static void main(String[] args) {  
        VarPrint t1 = new VarPrint();  
        t1.x = 88;  
        t1.y = 99;  
        int z1 = t1.x + t1.y;  
        VarPrint t2 = new VarPrint();  
        System.out.println(t2.x + " " + t2.y + " " + z1);  
    }  
}
```

- a. **30 99 178**
- b. **30 88 129**
- c. **30 99 187**
- d. **88 99 178**

Correct Answer:

- c. **30 99 187**

Detailed Solution:

If you perform any change for instance variable these changes won't be reflected for the remaining objects. Because for every object a separate copy of instance variable will be there. But if you do any change to the static variable, that change will be reflected for all objects because a static instance maintains a single copy in memory.

Please refer to chapter 3 of book Joy With Java for a more detailed explanation.



QUESTION 3:

What will be the output of the following Java program?

```
public class ArgumentTest {
    public static void main(String[] args) {
        Test t = new Test();
        t.start();
    }

    static class Test {
        void start() {
            int a = 4;
            int b = 5;
            System.out.print(" " + 8 + 3 + " ");
            System.out.print(a + b);
            System.out.print(" " + a + b + " ");
            System.out.print(foo() + a + b + " ");
            System.out.println(a + b + foo());
        }

        String foo() {
            return "foo";
        }
    }
}
```

- a. **9 7 7 foo34 34foo**
- b. **839 45foo45 9foo**
- c. **72 34 34 foo34 34foo**
- d. **9 7 7 foo 7 7foo**

Correct Answer:

- b. **839 45foo45 9foo**

Detailed Solution:

Here, print() methods internally converts the data in its argument into a String object and then print the composition. Here, + is the concatenation of different String representation.

Please refer to chapter 3 of book Joy With Java for a more detailed explanation.



QUESTION 4:

What is encapsulation in object-oriented programming?

- a. **Hiding implementation details and exposing only functionality**
- b. **The process of creating multiple objects in a program**
- c. **Writing multiple methods in a single class**
- d. **Using the this keyword to reference an object**

Correct Answer:

- a. **Hiding implementation details and exposing only functionality**

Detailed Solution:

Please refer to chapter 3 of book Joy With Java for a more detailed explanation.

Encapsulation is one of the fundamental principles of object-oriented programming. It involves bundling the data (variables) and the methods (functions) that operate on the data into a single unit, typically a class. By making certain data private and providing public methods to access or modify it, encapsulation helps hide implementation details from the outside world while exposing only the required functionality. This improves code modularity, security, and maintainability.



QUESTION 5:

Which of the following is true about constructors in a class?

- a. Constructors must have a return type.
- b. Constructors are used to initialize objects.
- c. A class can have only one constructor.
- d. Constructors cannot be overloaded.

Correct Answer:

- b. Constructors are used to initialize objects.

Detailed Solution:

A constructor is a special method in a class that is automatically called when an object of the class is created. Its main purpose is to initialize the object's properties (variables). Unlike other methods, constructors:

- Have the same name as the class.
- Do not have a return type, not even `void`. A class can have multiple constructors with different parameter lists (constructor overloading) to allow flexibility in object creation.

Please refer book **Joy with Java Chapter 3** for more detailed explanation.



QUESTION 6:

What does the `this` keyword in Java help to achieve?

- a. Avoiding name space collision between instance variables and method parameters
- b. Overloading methods in a class
- c. Accessing private methods in another class
- d. Creating multiple objects in a program

Correct Answer:

- a. Avoiding name space collision between instance variables and method parameters

Detailed Solution:

The `this` keyword is used to refer to the current instance of a class. It is commonly used to resolve ambiguity when instance variables and method parameters have the same name. For example:

```
class Example {  
    int value;  
  
    Example(int value) {  
        this.value = value; // Resolves name collision by referring to  
        // the instance variable  
    }  
}
```

Here, `this.value` refers to the instance variable, while `value` refers to the parameter of the constructor. This avoids confusion and ensures the proper assignment of values.
Please refer book Joy with Java Chapter 3 for more detailed explanation.



QUESTION 7:

What is the correct signature of the `main` method in Java?

- a. `public void main(String args[])`
- b. `public static void main(String args[])`
- c. `void main(String[] args)`
- d. `public static void main(String[] args)`

Correct Answer:

- d. `public static void main(String[] args)`

Detailed Solution:

The `main` method in Java must be declared as `public static void main(String[] args)` to be recognized by the JVM as the entry point of the program. The `public` modifier allows the method to be accessible from anywhere, `static` ensures it can be called without creating an instance of the class, and `String[] args` is the parameter used for command-line arguments.

Note: Please refer book Joy with Java Chapter 3 for more detailed explanation.



QUESTION 8:

Which of the following is used to take runtime input in Java?

- a. **BufferedReader**
- b. **Scanner**
- c. **DataInputStream**
- d. **All of the above**

Correct Answer:

- d. **All of the above**

Detailed Solution:

Java provides several ways to take runtime input:

BufferedReader: Reads text from the input stream efficiently.

Scanner: Parses primitive types and strings using regular expressions.

DataInputStream: Reads primitive data types and strings in a binary format. Each serves different purposes based on requirements.

Note: Please refer book Joy with Java Chapter 3 for more detailed explanation.



QUESTION 9:

Which method is used to format output in Java?

- a. **print()**
- b. **println()**
- c. **printf()**
- d. **format()**

Correct Answer:

- c. **printf()**

Detailed Solution:

The `printf()` method is used in Java to produce formatted output. It follows the syntax of `System.out.printf(format, arguments)` where the format specifies how the output should appear. For example:

```
System.out.printf("%d %s", 5, "Apples");
```

prints "5 Apples".

Note: Please refer book Joy with Java Chapter 3 for more detailed explanation.



QUESTION 10:

Which Java class is primarily used to read input from the console?

- a. **Scanner**
- b. **BufferedReader**
- c. **Console**
- d. **DataInputStream**

Correct Answer:

- a. **Scanner**

Detailed Solution:

The `Scanner` class is widely used to read input from the console in Java. It provides methods to parse and read primitive types (e.g., `nextInt()`, `nextDouble()`) and strings (e.g., `next()`, `nextLine()`). It is a versatile and easy-to-use class for input handling.

Note: Please refer to chapter 3 of book Joy With Java for a more detailed explanation.



PROGRAMMING IN JAVA

Assignment 03

TYPE OF QUESTION: MCQ

Number of questions: 10

Total marks: **10 × 1 = 10**

QUESTION 1:

Which of the following is true about the `super` keyword in Java?

- a. `super` can be used to call a parent class constructor.
- b. `super` is used to access private variables of the parent class.
- c. `super` is used to call a static method in the parent class.
- d. `super` can only be used inside a static method.

Correct Answer:

- a. `super` can be used to call a parent class constructor.

Detailed Solution:

The `super` keyword is used to refer to the immediate parent class. It is commonly used to invoke the parent class constructor or access its non-private methods and variables. However, it cannot access private members of the parent class or be used in static methods.



QUESTION 2:

What is the output of the following Java program?

```
class StaticScopeDemo {  
    static int x = 5;  
  
    public static void main(String[] args) {  
        int x = 10;  
        {  
            int x = 15; // Compilation Error  
            System.out.println(x);  
        }  
    }  
}
```

- a. 15**
- b. Compilation Error**
- c. 5**
- d. 10**

Correct Answer:

- b. Compilation Error**

Detailed Solution:

The block within `main()` tries to declare a local variable `x` that has the same name as an already existing variable in the same scope, which causes a compilation error. Variable names must be unique within the same scope



QUESTION 3:

What will be the output of the following program?

```
class Parent {  
    void display() {  
        System.out.println("Parent display");  
    }  
}  
  
class Child extends Parent {  
    void display() {  
        System.out.println("Child display");  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Parent obj = new Child();  
        obj.display();  
    }  
}
```

- a. Parent display**
- b. Child display**
- c. Compilation Error**
- d. Runtime Error**

Correct Answer:

- b. Child display**

Detailed Solution:

This is an example of **method overriding**. The `display()` method in the `Child` class overrides the method in the `Parent` class. When the method is called, Java uses dynamic method dispatch to execute the `Child` class implementation.



QUESTION 4:

Which of the following statements about abstract classes in Java is correct?

- a. Abstract classes can be instantiated directly.
- b. An abstract class must contain at least one abstract method.
- c. A class inheriting from an abstract class must implement all its abstract methods unless it is itself abstract.
- d. Abstract classes can be marked as `final`.

Correct Answer:

- c. A class inheriting from an abstract class must implement all its abstract methods unless it is itself abstract.

Detailed Solution:

Abstract classes cannot be instantiated directly. They may or may not contain abstract methods. A subclass inheriting an abstract class must provide implementations for all abstract methods unless it is also declared abstract.



QUESTION 5:

What will be the output of the following Java program?

```
public class NptelExample {  
    public static int fun(int n) {  
        if (n == 0) {  
            return 1;  
        }  
        return n * fun(n - 1);  
    }  
  
    public static void main(String[] args) {  
        System.out.println(fun(5));  
    }  
}
```

- a. 5
- b. 24
- c. 120
- d. Runtime Error

Correct Answer:

- c. 120

Detailed Solution:

The fun method is a recursive function that calculates the factorial of a number. For $\text{fun}(5)$, the computation is $5 * 4 * 3 * 2 * 1 = 120$.



QUESTION 6:

Which of the following is NOT true regarding the final keyword in Java?

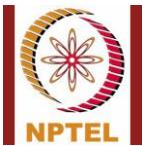
- a. A final method cannot be overridden in a subclass.
- b. A final variable can only be assigned once.
- c. A final class can have subclasses.
- d. A final variable can be assigned during declaration or in the constructor.

Correct Answer:

- c. A final class can have subclasses.

Detailed Solution:

A class marked as final cannot be extended, meaning it cannot have subclasses. The other statements about the final keyword are correct.



QUESTION 7:

What is the output of the following Java program?

```
class Test {  
    static int count = 0;  
  
    public Test() {  
        count++;  
    }  
  
    public static void main(String[] args) {  
        Test obj1 = new Test();  
        Test obj2 = new Test();  
        Test obj3 = new Test();  
        System.out.println("Count: " + Test.count);  
    }  
}
```

- a. Count: 0
- b. Compilation Error
- c. Runtime Error
- d. Count: 3

Correct Answer:

- d. Count: 3

Detailed Solution:

The `count` variable is static, meaning it is shared among all instances of the class. Each time a `Test` object is created, the constructor increments `count`. Since three objects are created, the output is `Count: 3`.



QUESTION 8:

Which of these is NOT an example of method overriding in Java?

- a. A subclass defining a method with the same name but different parameters than a superclass method.
- b. A subclass providing a new implementation for a method in the superclass.
- c. A subclass defining a method with the same name and parameters as a superclass method.
- d. Using the super keyword to call the superclass version of an overridden method.

Correct Answer:

- a. A subclass defining a method with the same name but different parameters than a superclass method.

Detailed Solution:

If a method in a subclass has the same name but different parameters, it is method overloading, not method overriding. Method overriding requires the same name and parameters.



QUESTION 9:

What is the output of the following Java program?

```
class Parent {  
    String message() {  
        return "Parent";  
    }  
}  
  
class Child extends Parent {  
    String message() {  
        return "Child";  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Child p = new Parent();  
        System.out.println(p.message());  
    }  
}
```

- a. Parent**
- b. Child**
- c. Compilation Error**
- d. No error and nothing is printed**

Correct Answer:

- c. Compilation Error**

Detailed Solution:

There will be a compilation error: Type mismatch: cannot convert from Parent to Child



QUESTION 10:

What is the output of the following program?

```
public class Nptel {  
    public static int fun(int n) {  
        if (n == 0) {  
            return 0;  
        }  
        return n + fun(n - 1);  
    }  
  
    public static void main(String[] args) {  
        System.out.println(fun(5));  
    }  
}
```

- a. 5
- b. 10
- c. 15
- d. Runtime Error

Correct Answer:

- c. 15

Detailed Solution:

The fun function calculates the sum of the first n natural numbers using recursion. For fun(5), the computation is $5 + 4 + 3 + 2 + 1 = 15$.



PROGRAMMING IN JAVA

Assignment 4

TYPE OF QUESTION: MCQ

Number of questions: 10

Total marks: $10 \times 1 = 10$

QUESTION 1:

Which access modifier allows a method to be accessible within the same package but not from outside the package?

- a. default
- b. private
- c. public
- d. protected

Correct Answer:

- a. default

Detailed Solution:

The default access modifier (no explicit modifier specified) allows access to the method within the same package but not from outside. It is also known as "package-private."



QUESTION 2:

What will happen if you attempt to access a private method from another class in Java?

- a. The method will be accessible.
- b. The method will throw an exception.
- c. The method will be inaccessible.
- d. The method will be converted to protected.

Correct Answer:

- c. The method will be inaccessible

Detailed Solution:

A private method is accessible only within the same class. Any attempt to access it from another class results in a compilation error.



QUESTION 3:

What will be the output of the following code?

```
class Person {  
    int a = 1;  
    int b = 0;  
    public Person() {  
        System.out.println(a + b + " Java ");  
    }  
}  
  
class Employee extends Person {  
    int a = 0;  
    int b = 1;  
    public Employee() {  
        System.out.println(a * b + " Java ");  
    }  
}  
  
public class Question {  
    public static void main(String args[]) {  
        Person p = new Employee();  
    }  
}
```

- a. 1 Java 10
0 Java 0**
- b. 1 Java
0 Java**
- c. 10 Java
0 Java**
- d. 1 java 1
0 Java 0**

Correct Answer:

- b. 1 Java
0 Java**

Detailed Solution:



If no super() or this() is included explicitly within the derived class constructor, the super() is implicitly invoked by the compiler. Therefore, in this case, the Person class constructor is called first and then the Employee class constructor is called. Up casting is allowed.



QUESTION 4:

Which of the following is a built-in Java package?

- a. `java.util`
- b. `my.package.util`
- c. `default.package`
- d. `system.java`

Correct Answer:

- a. `java.util`

Detailed Solution:

`java.util` is a built-in Java package containing utility classes such as `ArrayList`, `HashMap`, and `Scanner`.



QUESTION 5:

What keyword is used to define a package in Java?

- a. define
- b. package
- c. import
- d. namespace

Correct Answer:

- b. package

Detailed Solution:

The `package` keyword is used in Java to define a package. For example, `package mypackage;` defines a package named `mypackage`.



QUESTION 6:

How do you import a specific class from a package in Java?

- a. `import package.*;`
- b. `import package.ClassName;`
- c. `include package.ClassName;`
- d. `use package.ClassName;`

Correct Answer:

- b. `import package.ClassName;`

Detailed Solution:

The syntax `'import package.ClassName;'` imports a specific class from a package. For example, `'import java.util.Scanner;'` imports the `'Scanner'` class from the `'java.util'` package.



QUESTION 7:

Consider the following program:

```
class Base {  
    public void print() {  
        System.out.println("Base class...");  
    }  
}  
class Derived extends Base {  
    public void print() {  
        System.out.println("Derived class...");  
    }  
}  
public class Main {  
    private static void main (String[] args) {  
        Base b = new Base();  
        b.print();  
        Derived d = new Derived();  
        d.print();  
    }  
}
```

How many errors does this program contain?

- a. None
- b. 1
- c. 2
- d. 3

Correct Answer:

- b. 1

Detailed Solution:



This code has one error:

1. Incorrect visibility of `main` method:

The `main` method is defined as `private static void main(String[] args)`. The `main` method must be `public static void main(String[] args)` to be recognized as the entry point of the program by the Java Virtual Machine (JVM).



QUESTION 8:

Which of the following is true about Java interfaces?

- a. They cannot contain method implementations.
- b. An interface can extend multiple classes.
- c. An interface can only contain abstract methods.
- d. They allow a class to achieve multiple inheritance.

Correct Answer:

- d. They allow a class to achieve multiple inheritance.

Detailed Solution:

Interfaces in Java provide a mechanism to achieve multiple inheritance because a class can implement multiple interfaces.



QUESTION 9:

What will happen if you do not specify an access modifier for a class in a package?

- a. The class will be `private` by default.
- b. The class will be `protected` by default.
- c. The class will be accessible only within the same package.
- d. The class will be `public` by default.

Correct Answer:

- c. The class will be accessible only within the same package.

Detailed Solution:

If no access modifier is specified for a class, it gets the default access level, making it accessible only within the same package.



QUESTION 10:

Which access modifier provides the most restrictive access in Java?

- a. default
- b. protected
- c. private
- d. public

Correct Answer:

- c. private

Detailed Solution:

The `private` modifier provides the most restrictive access, allowing access only within the class where it is defined.



PROGRAMMING IN JAVA

Assignment 05

TYPE OF QUESTION: MCQ

Number of questions: 10

Total marks: **10 × 1 = 10**

QUESTION 1:

Which of the following statement(s) is/are true about `finally` in Java?

- I. The `finally` block is executed regardless of whether an exception is thrown or not.
- II. A `finally` block can exist without a `catch` block.
- III. The `finally` block will not execute if `System.exit()` is called in the `try` block.
- IV. A `finally` block can have a `return` statement, but it is not recommended to use.
 - a. I and II
 - b. II and III
 - c. I, II and III
 - d. I, II, III and IV

Correct Answer:

- d. I, II, III and IV

Detailed Solution:

The `finally` block always executes except when the JVM exits using `System.exit()`. It can exist without a `catch` block and may contain a `return` statement, though this is not recommended.



QUESTION 2:

What will be the output of the following Java program?

```
interface A {  
    int x = 10;  
    void display();  
}  
class B implements A {  
    public void display() {  
        System.out.println("Value of x: " + x);  
    }  
}  
public class Main {  
    public static void main(String[] args) {  
        B obj = new B();  
        obj.display();  
    }  
}
```

- a. Value of x: 10**
- b. Value of x: 0**
- c. Compilation Error**
- d. Runtime Error**

Correct Answer:

- a. Value of x: 10**

Detailed Solution:

Variables in interfaces are `public`, `static`, and `final` by default. Hence, the value of `x` is accessible in the `display` method.



QUESTION 3:

What will be the output of the following program?

```
class NPTEL {
    public static void main(String[] args) {
        try {
            int a = 5;
            int b = 0;
            System.out.println(a / b);
        } catch (ArithmException e) {
            System.out.print("Error ");
        } finally {
            System.out.print("Complete");
        }
    }
}
```

- a. 5 Complete**
- b. Error Complete**
- c. Runtime Error**
- d. Compilation Error**

Correct Answer:

- b. Error Complete**

Detailed Solution:

An `ArithmException` is caught in the `catch` block, which prints “Error”. The `finally` block executes afterward, printing “Complete”.



QUESTION 4:

Which of the following is TRUE regarding abstract class and an interface in Java?

- I. Abstract classes can contain constructors, but interfaces cannot.
 - II. Interfaces support multiple inheritance, but abstract classes do not.
 - III. Abstract classes can have both abstract and concrete methods, whereas interfaces only had abstract methods before Java 8.
-
- a. I, II and III
 - b. II only
 - c. I and II only
 - d. II and III only

Correct Answer:

- a. I, II and III

Detailed Solution:

Abstract classes can have constructors and concrete methods. Interfaces support multiple inheritance. Before Java 8, interfaces could only contain abstract methods, but now they can include default and static methods.



QUESTION 5:

Which of the following is a *checked exception* in Java?

- a. **NullPointerException**
- b. **ArrayIndexOutOfBoundsException**
- c. **IOException**
- d. **ArithmetricException**

Correct Answer:

- c. **IOException**

Detailed Solution:

IOException is a checked exception, meaning it must be either caught or declared in the throws clause of a method. The others are unchecked exceptions, which do not require explicit handling.



QUESTION 6:

Which keyword is NOT used by Java during exception handling?

- a. try
- b. catch
- c. final
- d. finally

Correct Answer:

- c. final

Detailed Solution:

In Java, exceptions are handled using the `try`, `catch`, and `finally` blocks. The `try` block contains code that might throw an exception, the `catch` block handles specific exceptions, and the `finally` block executes regardless of whether an exception occurs.



QUESTION 7:

What is the purpose of the `throws` keyword in Java?

- a. To declare exceptions that a method can throw
- b. To throw an exception immediately
- c. To catch an exception
- d. It is not a keyword in Java

Correct Answer:

- a. To declare exceptions that a method can throw

Detailed Solution:

The `throws` keyword is used in a method signature to declare the exceptions that the method might throw, alerting callers to handle or propagate these exceptions.



QUESTION 8:

Which of the following is TRUE about interfaces in Java?

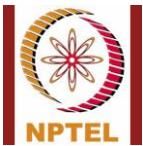
- a. Interfaces should always be defined as final
- b. Interfaces can be instantiated directly.
- c. Interfaces can extend multiple interfaces.
- d. Interfaces cannot have any methods signatures

Correct Answer:

- c. Interfaces can extend multiple interfaces.

Detailed Solution:

In Java, an interface can extend multiple interfaces. Interfaces can also contain public, static, and final variables, but they cannot be instantiated directly.



QUESTION 9:

What will be the output of the following code?

```
interface Demo {  
    void display();  
}  
  
class Test implements Demo {  
    public void display() {  
        System.out.println("Hello, NPTEL!");  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Test obj = new Test();  
        obj.display();  
    }  
}
```

- a. Hello, NPTEL!**
- b. Compilation Error**
- c. Runtime Error**
- d. No Output**

Correct Answer:

- a. Hello, NPTEL!**

Detailed Solution:

The Test class implements the Demo interface and provides a definition for the display method. When display() is called, it prints “Hello, NPTEL!”.



QUESTION 10:

What will be the output of the following Java program?

```
interface Calculator {  
    void calculate(int value);  
}  
  
class Square implements Calculator {  
    int result;  
    public void calculate(int value) {  
        result = value * value;  
        System.out.print("Square: " + result + " ");  
    }  
}  
  
class Cube extends Square {  
    public void calculate(int value) {  
        result = value * value * value;  
        super.calculate(value);  
        System.out.print("Cube: " + result + " ");  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Calculator obj = new Cube();  
        obj.calculate(3);  
    }  
}
```

- a. Square: 9 Cube: 9
- b. Cube: 27 Square: 9
- c. Square: 9 Square: 27 Cube: 27
- d. Square: 9 Cube: 27 Square: 27

Correct Answer:

- a. Square: 9 Cube: 9

Detailed Solution:

The `Cube` class overrides the `calculate` method of the `Square` class. In the `Cube` class's `calculate` method, `super.calculate(value)` is called, which executes the `calculate` method of the `Square`



NPTEL Online Certification Courses

Indian Institute of Technology Kharagpur



class. First, "Square: 9" is printed by the superclass method. Then, the overridden method in `Cube` prints "Cube: 9".



PROGRAMMING IN JAVA

Assignment 06

TYPE OF QUESTION: MCQ

Number of questions: 10

Total marks: $10 \times 1 = 10$

QUESTION 1:

Consider the following code snippet.

```
class MyThread extends Thread {  
    public void run() {  
        for (int i = 0; i < 3; i++) {  
            System.out.println("Running thread: " + i);  
        }  
    }  
  
    public class Main {  
        public static void main(String[] args) {  
            MyThread thread = new MyThread();  
            thread.run();  
            System.out.println("Main method complete.");  
        }  
    }  
}
```

What will be the output of the program?

- a. Running thread: 0
Running thread: 1
Running thread: 2
Main method complete.
- b. Running thread: 0
Main method complete.
Running thread: 1
Running thread: 2
- c. Main method complete.
- d. Error: The thread was not started using `start()`

Correct Answer:



- a. Running thread: 0
- Running thread: 1
- Running thread: 2
- Main method complete

Detailed Solution:

In this program, the `run()` method is called directly instead of using `start()`. When `run()` is called like this, it does not create a new thread; it behaves like a regular method call and runs on the main thread. As a result, the output is sequential. To create a separate thread, you must call the `start()` method. For a more detailed understanding of thread methods like `run()` and `start()`, refer to the book *Joy with Java*.



QUESTION 2:

Which of the following best describes the concept of multithreading in Java?

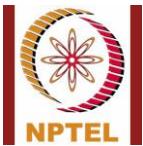
- a. Multiple threads execute concurrently, sharing the same memory space.
- b. Only one thread executes at a time, ensuring sequential execution.
- c. Threads in Java cannot communicate with each other.
- d. Threads require separate memory allocation for each thread to run.

Correct Answer:

- a. Multiple threads execute concurrently, sharing the same memory space.

Detailed Solution:

Multithreading in Java allows multiple threads to run concurrently within the same program, sharing the same memory space. This feature makes Java programs more efficient, especially for tasks like multitasking or background processing. However, proper synchronization is crucial to avoid issues like data inconsistency. For a more detailed explanation of multithreading concepts, refer to the book *Joy with Java*.



QUESTION 3:

What will happen when the following code is executed?

```
class ExampleThread extends Thread {  
    public void run() {  
        System.out.println("Thread is running.");  
    }  
  
    public class Main {  
        public static void main(String[] args) {  
            ExampleThread thread = new ExampleThread();  
            thread.start();  
            thread.start();  
        }  
    }  
}
```

- a. The program will execute successfully, printing "Thread is running." twice.
- b. The program will throw an error when attempting to start the thread a second time.
- c. The program will terminate without any output.
- d. The thread will run only once, and the second start() call will be ignored.

Correct Answer:

- b. The program will throw an error when attempting to start the thread a second time.**

Detailed Solution:

In Java, a thread can only be started once. If you try to call start() on the same thread object more than once, the JVM will throw an IllegalThreadStateException. This is because a thread's lifecycle allows it to transition to the "terminated" state after completing execution, and it cannot be restarted.

For more details about the thread lifecycle, refer to the book *Joy with Java*.



QUESTION 4:

Find the error in the following program:

```
class RunnableExample implements Runnable {  
    public void run() {  
        for (int i = 1; i <= 3; i++) {  
            System.out.println("Runnable thread: " + i);  
        }  
    }  
  
    public class Main {  
        public static void main(String[] args) {  
            RunnableExample runnable = new RunnableExample();  
            runnable.run();  
            System.out.println("Main method ends.");  
        }  
    }  
}
```

- a. The program will throw an error because Runnable cannot be executed directly.
- b. The program will run successfully but will not create a new thread.
- c. The program will create a new thread and execute concurrently.
- d. The program will throw a runtime error because Thread is not used.

Correct Answer:

- b. The program will run successfully but will not create a new thread.

Detailed Solution:

In this code, the Runnable object is created, but it is executed directly using the run() method instead of passing it to a Thread object. To create a new thread, you need to use:

```
Thread t = new Thread(runnable);  
t.start();
```

Without this, the run() method behaves like a regular method call, and no new thread is created. For more detailed guidance on creating threads using Runnable, refer to the book Joy with Java.



QUESTION 5:

What will happen when the following code is executed?

```
class RunnableExample implements Runnable {  
    public void run() {  
        for (int i = 1; i <= 3; i++) {  
            System.out.println("Thread running: " + i);  
        }  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        RunnableExample task = new RunnableExample();  
        Thread thread = new Thread(task);  
        thread.start();  
    }  
}
```

- a. The program will throw a compile-time error because `Runnable` is not a thread.
- b. The program will execute successfully, and the `run()` method will run in a new thread.
- c. The program will execute the `run()` method directly on the main thread.
- d. The program will throw a runtime error because `Runnable` is not properly implemented.

Correct Answer:

- b. The program will execute successfully, and the `run()` method will run in a new thread.

Detailed Solution:

The Runnable interface is implemented by the class `RunnableExample`. To create a thread, the `Runnable` object is passed to the `Thread` constructor, and calling `start()` ensures that the `run()` method is executed in a new thread. For more details on creating threads using `Runnable`, refer to the book *Joy with Java*.



QUESTION 6:

Which of the following states can a thread enter during its lifecycle in Java?

- a. New, Runnable, Running, Blocked
- b. New, Runnable, Waiting, Blocked, Terminated
- c. New, Runnable, Running, Sleeping, Dead
- d. New, Active, Waiting, Suspended, Terminated

Correct Answer:

- b. New, Runnable, Waiting, Blocked, Terminated

Detailed Solution:

The states of a thread in Java are:

- **New:** Thread is created but not started.
- **Runnable:** Thread is ready to run but waiting for CPU time.
- **Waiting/Blocked:** Thread is paused or waiting for some resource.
- **Terminated:** Thread has completed execution.

For a detailed explanation of thread states, refer to the book *Joy with Java*.



QUESTION 7:

What does the thread scheduler use to decide which thread to run when multiple threads are in the runnable state?

- a. Thread priority
- b. Thread's execution time
- c. Thread name
- d. Thread creation order

Correct Answer:

- a. Thread priority

Detailed Solution:

The thread scheduler uses thread priorities as a hint to determine the execution order of threads in the runnable state. However, thread scheduling also depends on the operating system's scheduling policies and may not strictly follow priority.

For more on thread scheduling, refer to the book *Joy with Java*.



QUESTION 8:

Consider the following program:

```
class PriorityExample extends Thread {  
    public void run() {  
        System.out.println(Thread.currentThread().getName() +  
                           " with priority " +  
                           Thread.currentThread().getPriority());  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        PriorityExample t1 = new PriorityExample();  
        PriorityExample t2 = new PriorityExample();  
  
        t1.setPriority(Thread.MIN_PRIORITY);  
        t2.setPriority(Thread.MAX_PRIORITY);  
  
        t1.start();  
        t2.start();  
    }  
}
```

Which of the following is true about the output?

- a. The thread with the higher priority is guaranteed to execute first.
- b. The thread with the lower priority will never execute.
- c. The order of execution depends on the JVM and OS scheduling policies.
- d. The program will throw an error due to invalid priority values.

Correct Answer:

- c. The order of execution depends on the JVM and OS scheduling policies.

Detailed Solution:

Thread priority is a hint to the thread scheduler but does not guarantee execution order. The actual behavior depends on the JVM and the underlying OS. For more on thread priorities, refer to the book *Joy with Java*.



QUESTION 9:

What is the primary purpose of thread synchronization in Java?

- a. To allow multiple threads to execute a method at the same time
- b. To ensure thread execution follows a specific order
- c. To prevent race conditions and ensure data consistency
- d. To allow threads to communicate with each other

Correct Answer:

- c. To prevent race conditions and ensure data consistency

Detailed Solution:

Thread synchronization is used to control the access of multiple threads to shared resources, ensuring data consistency and preventing race conditions. This is typically done using synchronized methods or blocks. For more details on thread synchronization, refer to the book *Joy with Java*.



QUESTION 10:

What is the primary difference between Byte Streams and Character Streams in Java?

- a. Byte Streams handle characters, while Character Streams handle bytes.
- b. Byte Streams are used for binary data, while Character Streams are used for text data.
- c. Character Streams are faster than Byte Streams in all cases.
- d. Character Streams cannot handle international characters like Unicode.

Correct Answer:

- b. Byte Streams are used for binary data, while Character Streams are used for text data.

Detailed Solution:

Byte Streams: Handle raw binary data like images or files (InputStream, OutputStream).

Character Streams: Handle text data and support encoding like Unicode (Reader, Writer).

Character Streams are better for text processing, especially when handling international characters.

For a detailed understanding of Java I/O streams, refer to the book *Joy with Java*.



PROGRAMMING IN JAVA

Assignment 7

TYPE OF QUESTION: MCQ

Number of questions: 10

Total marks: $10 \times 1 = 10$

QUESTION 1:

What will be the output of the following Java program?

```
import java.io.*;
class ReadFile {
    public static void main(String[] args) throws IOException {
        FileReader fr = new FileReader("NPTEL.txt");
        BufferedReader br = new BufferedReader(fr);
        System.out.println(br.readLine());
        br.close();
    }
}
```

Assume NPTEL.txt contains:

This is Programming in Java online course.

- a. Hello, World!**
- b. This is Programming in Java online course.**
- c. IOException**
- d. null**

Correct Answer:

- b. This is Programming in Java online course.**

Detailed Solution:

BufferedReader to read the first line from the file NPTEL.txt. Since the file contains "This is Programming in Java online course.", it is printed.



QUESTION 2:

Which of these classes is used to write primitive data types to an output stream in Java?

- a. **FileWriter**
- b. **DataOutputStream**
- c. **PrintWriter**
- d. **BufferedOutputStream**

Correct Answer:

- b. **DataOutputStream**

Detailed Solution:

The `DataOutputStream` class is used to write primitive data types (e.g., `int`, `float`, `boolean`) to an output stream.



QUESTION 3:

What will the following code print?

```
import java.io.*;

class RandomAccessFileExample {
    public static void main(String[] args) throws IOException {
        RandomAccessFile file = new RandomAccessFile("test.dat", "rw");
        file.writeInt(100);
        file.seek(0);
        System.out.println(file.readInt());
        file.close();
    }
}
```

- a. 0**
- b. Runtime Error**
- c. Compilation Error**
- d. 100**

Correct Answer:

- d. 100**

Detailed Solution:

The program writes the integer 100 to the file test.dat. Using the seek(0) method, the file pointer is reset to the beginning, and the integer is read back and printed.



QUESTION 4:

Complete the following snippet with the required code.

```
File file = new File("file.txt");
if (_____ ) // Fill in the blanks
{
    System.out.println("File exists.");
} else
{
    System.out.println("File does not exist.");
}
```

- a. `file.exists()`
- b. `file.isFile()`
- c. `file.fileExists()`
- d. `file.isAvailable()`

Correct Answer:

- a. `file.exists()`

Detailed Solution:

The `exists()` method in the `File` class checks if a file or directory exists in the specified path.



QUESTION 5:

What will the following Java program output?

```
import java.io.*;  
  
class SequenceInputStreamExample {  
    public static void main(String[] args) throws IOException {  
        ByteArrayInputStream input1 = new  
ByteArrayInputStream("123".getBytes());  
        ByteArrayInputStream input2 = new  
ByteArrayInputStream("ABC".getBytes());  
        SequenceInputStream sequence = new  
SequenceInputStream(input1, input2);  
  
        int i;  
        while ((i = sequence.read()) != -1) {  
            System.out.print((char) i);  
        }  
    }  
}
```

- a. 123ABC**
- b. ABC123**
- c. Compilation Error**
- d. Runtime Error**

Correct Answer:

- a. 123ABC**

Detailed Solution:

The `SequenceInputStream` combines `input1` and `input2` streams sequentially. It first reads the content of `input1` (123), followed by `input2` (ABC), resulting in 123ABC.



QUESTION 6:

What is the output of the following Java program?

```
class Main {  
  
    public static void main(String args[]) {  
        final int i;  
        i = 20;  
        i = 30;  
        System.out.println(i);  
    }  
}
```

- a. 30
- b. Compiler Error
- c. Garbage value
- d. 0

Correct Answer:

- b. Compiler Error

Detailed Solution:

i is assigned a value twice. Final variables can be assigned values only one. Following is the compiler error “Main.java:5: error: variable i might already have been assigned”.



QUESTION 7:

What will be the output of the following Java program?

```
import java.io.*;
class Chararrayinput {
    public static void main(String[] args) {
        String obj = "abcdefgh";
        int length = obj.length();
        char c[] = new char[length];
        obj.getChars(0, length, c, 0);
        CharArrayReader input1 = new CharArrayReader(c);
        CharArrayReader input2 = new CharArrayReader(c, 1, 4);
        int i, j;
        try {
            while ((i = input1.read()) == (j = input2.read())) {
                System.out.print((char) i);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

- a. abc
- b. abcd
- c. abcde
- d. none of the mentioned

Correct Answer:

- d. none of the mentioned

Detailed Solution:

No output is printed. CharArrayReader object input1 contains string “abcdefgh” whereas object input2 contains string “bcde”, when while((i=input1.read())==(j=input2.read())) is executed the starting character of each object is compared since they are unequal control comes out of loop and nothing is printed on the screen.



QUESTION 8:

What will be the output of the following Java program?

```
public class Calculator {
    int num = 100;

    public void calc(int num) {
        this.num = num * 10;
    }

    public void printNum() {
        System.out.println(num);
    }

    public static void main(String[] args) {
        Calculator obj = new Calculator();
        obj.calc(2);
        obj.printNum();
    }
}
```

- a. 20
- b. 100
- c. 1000
- d. 2

Correct Answer:

- a. 20

Detailed Solution:

Here the class instance variable name(num) is same as calc() method local variable name(num). So for referencing class instance variable from calc() method, this keyword is used. So in statement `this.num = num * 10`, num represents local variable of the method whose value is 2 and this.num represents class instance variable whose initial value is 100. Now in printNum() method, as it has no local variable whose name is same as class instance variable, so we can directly use num to reference instance variable, although this.num can be used.



QUESTION 9:

What will be the output of the following code?

```
import java.io.*;  
  
public class W7 {  
    public static void main(String[] args) {  
        try {  
  
            PrintWriter writer = new PrintWriter(System.out);  
  
            writer.write(9 + 97);  
  
            writer.close();  
        } catch (Exception e) {  
            System.out.println(e);  
        }  
    }  
}
```

- a. It will give compile-time error
- b. It will give run-time error
- c. j
- d. 106

Correct Answer:

- c. j

Detailed Solution:

The output of this program will be the character 'j' because the Unicode code point for 106 corresponds to 'j'.



QUESTION 10:

What will be the output of the following code.

file.txt contain "This is Programming in Java online course." (without quotes)

```
import java.io.File;

class FileSizeExample {
    public static void main(String[] args) {
        // Specify the file path
        String filePath = "file.txt";

        // Create a File object
        File file = new File(filePath);

        // Get the size of the file
        long fileSize = file.length();

        // Print the size of the file
        System.out.println(fileSize);
    }
}
```

- a. 42**
- b. 35**
- c. 7**
- d. 0**

Correct Answer:

- a. 42**

Detailed Solution:

The length() method on the File object, which returns the size of the file in bytes.



PROGRAMMING IN JAVA

Assignment 8

TYPE OF QUESTION: MCQ

Number of questions: 10

Total marks: $10 \times 1 = 10$

QUESTION 1:

Which of the following is TRUE regarding check box and radio button?

- a. Check box is used for single selection item whereas radio button is used for multiple selection.
- b. Check box is used for multiple selection items whereas radio button is used for single selection.
- c. Both are used for multiple as well as single item selection.
- d. Checkbox is always preferred than radio buttons.

Correct Answer:

- b. Check box is used for multiple selection items whereas radio button is used for single selection.

Detailed Solution:

Check box is used for multiple selection items whereas radio button is used for single selection. For example, if a form is asking for your favorite hobbies, there might be multiple correct answers to it, in that case check box is preferred. And if a form is asking about gender, there must be only one true option among the multiple choices, in that case radio buttons are used.



QUESTION 2:

Which of the following is the latest graphics and media package for Java?

- a. Applet
- b. AWT
- c. Swing
- d. JavaFX

Correct Answer:

- d. JavaFX

Detailed Solution:

JavaFX is a set of latest graphics and media packages in Java that enables developers to design, create, test, debug, and deploy rich client applications that operate consistently across diverse platforms.

More details can be found here: <https://docs.oracle.com/javafx/2/overview/fxpub-overview.htm>



QUESTION 3:

Which of the following is/are NOT an exception of java.awt package?

- a. HeadlessException
- b. AWTException
- c. FontFormatException
- d. IllegalStateException

Correct Answer:

- d. IllegalStateException

Detailed Solution:

IllegalStateException does not belong to java.awt package.



QUESTION 4:

Which of the following statement is FALSE about the update() in java.awt package?

- a. Sets the color of the graphics context to be the foreground color of this component.
- b. Calls this component's paint method to completely redraw this component.
- c. Updates the component by checking an online repository.
- d. Clears this component by filling it with the background color.

Correct Answer:

- c. Updates the component by checking an online repository

Detailed Solution:

The update() function does not update the component by checking an online repository rather it, sets the color of the graphics context to be the foreground color of this component, calls this component's paint method to completely redraw this component and clears this component by filling it with the background color.



QUESTION 5:

Which of the following method is used to remove all items from scrolling list in java.awt.list?

- a. hide()
- b. remove()
- c. clear()
- d. close()

Correct Answer:

- c. clear()

Detailed Solution:

The function clear() in java.awt.list is used for remove all items from scrolling list.



QUESTION 6:

Which layout manager arranges components in a single row or column in Java?

- a. **FlowLayout**
- b. **BorderLayout**
- c. **GridLayout**
- d. **CardLayout**

Correct Answer:

- a. **FlowLayout**

Detailed Solution:

The **FlowLayout** arranges components sequentially, either in a single row or column, based on the available space.

Commented [NP1]: no question shold be given from applet

Commented [DM2R1]: changed. Thanks.



QUESTION 7:

What does "AWT" stand for in Java?

- a. Abstract Widget Toolkit
- b. Advanced Window Toolkit
- c. Abstract Window Toolkit
- d. Advanced Widget Toolkit

Correct Answer:

- c. Abstract Window Toolkit

Detailed Solution:

AWT is the Java library for creating GUI components like buttons and windows.



QUESTION 8:

Which AWT component is used to create a button in a GUI?

- a. Label
- b. Button
- c. TextField
- d. Checkbox

Correct Answer:

- b. Button

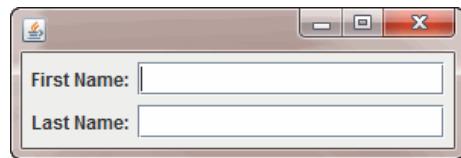
Detailed Solution:

The `Button` class in AWT is used to create clickable buttons in a GUI.



QUESTION 9:

Which of the following is TRUE about the following GUI?



- a. There is a Frame and two TextFields.
- b. There is a Frame with two Labels and two non-editable TextFields.
- c. There are two Labels.
- d. There is a Frame with two Labels and two TextFields.

Correct Answer:

- d. There is a Frame with two Labels and two TextFields

Detailed Solution:

There is a Frame with two Labels and two TextFields in the given GUI.



QUESTION 10:

Which of the following is TRUE about check box in Java?

- a. A check box can't be in either an "on" (true) or "off" (false) state.
- b. Clicking on a check box changes its state from "on" to "off," or from "off" to "on."
- c. A check box can be in an "on" (true) and in "off" (false) state simultaneously.
- d. Check boxes cannot be grouped together.

Correct Answer:

- b. Clicking on a check box changes its state from "on" to "off," or from "off" to "on."

Detailed Solution:

A check box is a graphical component that can be in either an "on" (true) or "off" (false) state. Clicking on a check box changes its state from "on" to "off," or from "off" to "on.". A check box cannot be in both "on" and "off" state simultaneously. Further, several check boxes can be grouped together under the control of a single object, using the CheckboxGroup class. In a check box group, at most one button can be in the "on" state at any given time. Clicking on a check box to turn it on forces any other check box in the same group that is on into the "off" state.

Commented [NP3]: Please only set mcq type question (one correct answer)

In FINAL QP you can set MSQ

Commented [DM4R3]: fixed



PROGRAMMING IN JAVA

Assignment X

TYPE OF QUESTION: MCQ

Number of questions: 10

Total marks: $10 \times 1 = 10$

QUESTION 1:

Which Swing component is best suited for displaying a drop-down list of selectable options?

- a. JButton
- b. JComboBox
- c. JTextField
- d. JPanel

Correct Answer:

- b. JComboBox

Detailed Solution:

JComboBox is a Swing component that provides a drop-down list from which users can select one option.



QUESTION 2:

What will be the output of the following Java code?

```
import javax.swing.*;
import java.awt.*;

public class SwingExample {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Example");
        frame.setLayout(new FlowLayout());
        frame.add(new JButton("Button 1"));
        frame.add(new JButton("Button 2"));
        frame.setSize(300, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

- a. A frame with two buttons labeled “Button 1” and “Button 2”.
- b. A frame with only one button labeled “Button 2”.
- c. Compilation Error.
- d. Runtime Error.

Correct Answer:

- a. A frame with two buttons labeled “Button 1” and “Button 2”.

Detailed Solution:

The `FlowLayout` layout manager arranges components in a row. Here, both buttons are added and displayed in the frame.



QUESTION 3:

Which of the following is true about the `JLabel` component in Java Swing?

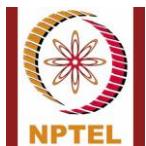
- a. It is used only for displaying text.
- b. It can display text and icons.
- c. It cannot be added to a `JPanel`.
- d. It generates mouse events by default.

Correct Answer:

- b. It can display text and icons.

Detailed Solution:

`JLabel` can display both text and images/icons. It is commonly used for non-interactive purposes in Swing GUIs.



QUESTION 4:

Which method is used to handle mouse click events in Java Swing?

- a. `mouseClicked()`
- b. `keyPressed()`
- c. `actionPerformed()`
- d. `componentShown()`

Correct Answer:

- a. `mouseClicked()`

Detailed Solution:

The `mouseClicked()` method in the `MouseListener` interface is used to handle mouse click events in Java Swing.



QUESTION 5:

What should replace // **INSERT CODE HERE** to create a **JFrame** with a **JButton** labeled "Click Me"?

```
import javax.swing.*;  
  
public class FrameExample {  
    public static void main(String[] args) {  
        JFrame frame = new JFrame("Demo Frame");  
        JButton button = new JButton("Click Me");  
        // INSERT CODE HERE  
        frame.setSize(300, 200);  
        frame.setVisible(true);  
    }  
}
```

- a. **frame.add(button);**
- b. **frame.insert(button);**
- c. **frame.append(button);**
- d. **frame.push(button);**

Correct Answer:

- a. **frame.add(button);**

Detailed Solution:

The **add()** method is used to add components like buttons, text fields, or panels to a **JFrame**.



QUESTION 6:

Identify and correct the error in the following program:

```
import javax.swing.*;
import java.awt.*;

public class PanelExample {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Panel Example");
        JPanel panel = new JPanel();
        JButton button = new JButton("Submit");
        panel.setLayout(); // ERROR
        panel.add(button);
        frame.add(panel);
        frame.setSize(300, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

What should the erroneous line (`panel.setLayout();`) be replaced with?

- a. `panel.setLayout(new GridLayout());`
- b. `panel.addFlowLayout();`
- c. `panel.appendLayout(new FlowLayout());`
- d. `panel.setLayout(new FlowLayout());`

Correct Answer:

- d. `panel.setLayout(new FlowLayout());`

Detailed Solution:

The `setLayout()` method is used to define the layout manager for a panel. `FlowLayout` is the default layout for `JPanel`.



QUESTION 7:

What will the following Java program output?

```
import javax.swing.*;  
  
public class LabelExample {  
    public static void main(String[] args) {  
        JFrame frame = new JFrame("Label Demo");  
        JLabel label = new JLabel("Welcome to Swing");  
        frame.setSize(250, 100);  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.setVisible(true);  
    }  
}
```

- a. A frame with the label "Welcome to Swing".
- b. A frame with no visible label.
- c. Compilation Error.
- d. Runtime Error.

Correct Answer:

- b. A frame with no visible label.

Detailed Solution:

The `JLabel` has to be added to the frame using `add()`. The frame is visible, but the label is not displayed as it has not been added.



QUESTION 8:

What does the following code do?

```
import javax.swing.*;  
  
public class NPTEL extends JFrame {  
    JButton button;  
  
    public NPTEL() {  
        button = new JButton("Programming in Java");  
        add(button);  
        setSize(300, 200);  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setVisible(true);  
    }  
  
    public static void main(String[] args) {  
        new NPTEL();  
    }  
}
```

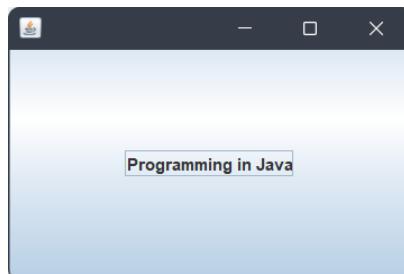
- a. Creates a JFrame with a JButton labeled "Programming in Java"
- b. Compiles with errors
- c. Displays a message dialog
- d. Creates a JPanel with a JButton labeled "Programming in Java"

Correct Answer:

- a. Creates a JFrame with a JButton labeled "Programming in Java"

Detailed Solution:

The code extends `JFrame` and uses the `JButton` class object to create a button with the name of "Programming in Java".





QUESTION 9:

What happens when the button in this Java code snippet is clicked?

```
import javax.swing.*;
import java.awt.event.*;
public class NPTEL {
    public static void main(String[] args) {
        JFrame frame = new JFrame("NPTEL Java Course");
        JButton button = new JButton("Click Me");
        button.setBounds(50, 100, 100, 40);
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                JOptionPane.showMessageDialog(null, "Welcome to the course");
            }
        });
        frame.add(button);
        frame.setSize(300, 200);
        frame.setLayout(null);
        frame.setVisible(true);
    }
}
```

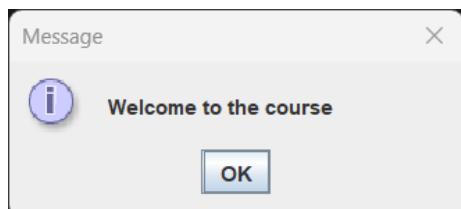
- a. The program exits**
- b. A message dialog with the text "Welcome to the course" is displayed**
- c. The button label changes to "Welcome to the course"**
- d. Nothing happens**

Correct Answer:

- b. A message dialog with the text "Welcome to the course" is displayed**

Detailed Solution:

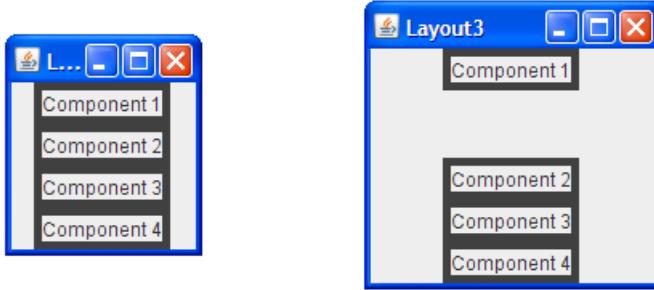
The code creates a button with label “Click Me” and in the frame titled “NPTEL Java Course”. A action listener is defined that opens a new message dialog with the text “Welcome to the course” when the button is clicked.





QUESTION 10:

The container displays a number of components in a column, with extra space going between the first two components.



Which of the following layout manager(s) most naturally suited for the described layout?

- a. BoxLayout
- b. FlowLayout
- c. BorderLayout
- d. GridLayout

Correct Answer:

- a. BoxLayout

Detailed Solution:

BoxLayout lays out components in either a column or a row. You can specify extra space using an invisible component.



PROGRAMMING IN JAVA

Assignment 10

TYPE OF QUESTION: MCQ

Number of questions: 10

Total marks: $10 \times 1 = 10$

QUESTION 1:

Which Swing component is used to create a button?

- a. JLabel
- b. JButton
- c. JTextField
- d. JPanel

Correct Answer:

- b. JButton

Detailed Solution:

JButton is used to create a clickable button in a Swing-based GUI.



QUESTION 2:

What does a toggle button in Java do?

- a. It switches between two states, like on/off or enabled/disabled.
- b. It only changes the color of the button when clicked.
- c. It displays text inside the button.
- d. It opens a new window when clicked.

Correct Answer:

- a. It switches between two states, like on/off or enabled/disabled.

Detailed Solution:

A toggle button allows users to switch between two states, such as "on" or "off." It behaves like a checkbox, but it is represented as a button.



QUESTION 3:

What is the primary purpose of SQL(Structured Query Language) in Java application ?

- a. To create and amange user interfaces
- b. To interact with databases for data retrieval and manipulation
- c. To performa systme-level operation like file management
- d. To handle memory allocation in Java programs

Correct Answer:

- b. To interact with databases for data retrieval and manipulation

Detailed Solution:

SQL is used in Java applications primarily to connect to databases, retrieve data, update records, and perform other database operations, making it essential for handling data within applications.

Commented [NP1]: Already given in W11, requesting to change the question

Commented [DM2R1]: Thanks , question changed.



QUESTION 4:

Which method is used in Java to check if a network interface is up and running?

- a. `isActive()`
- b. `isUp()`
- c. `isRunning()`
- d. `isConnected()`

Correct Answer:

- b. `isUp()`

Detailed Solution:

In Java, the `isUp()` method is used to check whether a network interface is up and running. This method is part of the `NetworkInterface` class from the `java.net` package.



QUESTION 5:

Which of the following are considered as basic networking components in Java?

- a. **Socket and ServerSocket**
- b. **String and Integer**
- c. **JFrame and JButton**
- d. **Scanner and System**

Correct Answer:

- a. **Socket and ServerSocket**

Detailed Solution:

In Java, **Socket** and **ServerSocket** are the primary components used for establishing client-server communication. These classes belong to the `java.net` package and are fundamental to networking in Java. **String**, **Integer**, **JFrame**, **JButton**, **Scanner**, and **System** are not used for basic networking functionality.



QUESTION 6:

In context of the following URL, identify the correct option.

<https://nptel.ac.in>

- a. There is no protocol provided in the above link.
- b. The website provides a secure connection.
- c. The given link is incomplete and hence cannot open a website.
- d. The ac.in refers to the domain.

Correct Answer:

- b. The website provides a secure connection.

Detailed Solution:

Resource name is nptel.ac.in. The protocol used is https and hence provides a secure connection to the website. The link is complete in all aspects and hence can open a website (*if hosted*). The ac.in portion of the website is top-level domain part and not a path.



QUESTION 7:

Which of the following is/are application layer protocol(s)?

- a. TCP
- b. UDP
- c. ARP
- d. SMTP

Correct Answer:

- d. SMTP

Detailed Solution:

TCP, UDP are transport layer protocols. ARP is a Network - IP layer protocol. SMTP is Application layer protocol.



QUESTION 8:

What does this Java code snippet do?

```
import java.sql.*;  
  
public class NPTEL {  
    public static void main(String[] args) {  
        try {  
            Class.forName("com.mysql.jdbc.Driver");  
            Connection con = DriverManager.getConnection(  
                "jdbc:mysql://localhost:3306/mydatabase",  
                "username", "password");  
            Statement stmt = con.createStatement();  
            ResultSet rs = stmt.executeQuery("SELECT * FROM employees");  
            while (rs.next()) {  
                System.out.println(rs.getString(1)+" "+rs.getString(2));  
            }  
            con.close();  
        } catch (Exception e) {  
            System.out.println(e);  
        }  
    }  
}
```

- a. Connects to a MySQL database, retrieves data from the "employees" table, and prints it
- b. Inserts data into the "employees" table of a MySQL database
- c. Deletes data from the "employees" table of a MySQL database
- d. Updates data in the "employees" table of a MySQL database

Correct Answer:

- a. Connects to a MySQL database, retrieves data from the "employees" table, and prints it

Detailed Solution:

The code snippet establishes a connection to a MySQL database, executes a SELECT query to retrieve data from the "employees" table, and prints the results.



QUESTION 9:

Which class provides methods to work with URLs?

- a. `URLConnection`
- b. `HttpURL`
- c. `NetURL`
- d. `URL`

Correct Answer:

- d. `URL`

Detailed Solution:

The `URL` class provides methods to work with Uniform Resource Locators.



QUESTION 10:

Which class in Java is used to create a server socket?

- a. **Socket**
- b. **ServerSocket**
- c. **DatagramSocket**
- d. **HttpURLConnection**

Correct Answer:

- b. ServerSocket**

Detailed Solution:

ServerSocket is used to listen for incoming connections on a specific port, enabling communication with clients in server-client applications.



PROGRAMMING IN JAVA

Assignment 11

TYPE OF QUESTION: MCQ

Number of questions: 10

Total marks: $10 \times 1 = 10$

QUESTION 1:

What is the full form of JDBC?

- a. Java Database Connectivity
- b. Java Data Code
- c. Java Data Communication
- d. Java Development Connectivity

Correct Answer:

- a. Java Database Connectivity

Detailed Solution:

JDBC stands for Java Database Connectivity, a Java API used to connect and interact with relational databases. It provides methods to query and update data in a database.



QUESTION 2:

Fill in the missing code to establish a connection to a MySQL database.

```
import java.sql.*;  
  
public class JDBCExample {  
    public static void main(String[] args) {  
        try {  
            // Load JDBC Driver  
            Class.forName("com.mysql.cj.jdbc.Driver");  
  
            // Establish Connection  
            Connection connection = // INSERT CODE HERE;  
  
            System.out.println("Connected to the database.");  
            connection.close();  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

What should replace // INSERT CODE HERE ?

- a. `DriverManager.connect("mysql:localhost:mydb", "user", "password");`
- b. `DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb", "user", "password");`
- c. `Connection.get("jdbc:mysql://localhost:3306/mydb", "user", "password");`
- d. `Driver.connect("jdbc:mysql://localhost:mydb", "user", "password");`

Correct Answer:

- b. `DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb", "user", "password");`

Detailed Solution:

The `DriverManager.getConnection` method establishes a connection to the specified database URL with the provided username and password.



QUESTION 3:

Identify the error in the following code and select the corrected statement:

```
import java.sql.*;

public class ResultSetExample {
    public static void main(String[] args) throws SQLException {
        Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb",
"user", "password");
        Statement stmt = connection.createStatement();
        ResultSet rs = stmt.execute("SELECT * FROM users"); // Error
        while (rs.next()) {
            System.out.println(rs.getString("username"));
        }
        connection.close();
    }
}
```

What is the correct statement to replace the line with error (`stmt.execute("SELECT * FROM users")`)?

- a. `ResultSet rs = stmt.executeQuery("SELECT * FROM users");`
- b. `ResultSet rs = stmt.runQuery("SELECT * FROM users");`
- c. `ResultSet rs = stmt.execute("users SELECT * FROM");`
- d. `ResultSet rs = stmt.fetch("SELECT * FROM users");`

Correct Answer:

- a. `ResultSet rs = stmt.executeQuery("SELECT * FROM users");`

Detailed Solution:

The `executeQuery` method is used to execute SQL queries that return a `ResultSet`.



QUESTION 4:

What will the following Java program output if the database contains a table products with a column name and three rows: Laptop, Phone, and Tablet?

```
import java.sql.*;

public class DisplayProducts {
    public static void main(String[] args) {
        try {
            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/store", "root",
"pass");
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT name FROM products");
            while (rs.next()) {
                System.out.println(rs.getString("name"));
            }
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

- a. **Compilation Error**
- b. **Runtime Error**
- c. **Laptop
Phone
Tablet**
- d. **No Output**

Correct Answer:

- c. **Laptop
Phone
Tablet**

Detailed Solution:

The program fetches and displays all rows from the `name` column in the `products` table using a `ResultSet`.



QUESTION 5:

Complete the following code to insert a new user into a users table.

```
import java.sql.*;  
  
public class InsertUser {  
    public static void main(String[] args) {  
        try {  
            Connection conn =  
DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb", "root",  
"password");  
            String query = "INSERT INTO users (username, email) VALUES  
(?, ?);  
            PreparedStatement pstmt = // INSERT CODE HERE;  
            pstmt.setString(1, "john_doe");  
            pstmt.setString(2, "john@example.com");  
            pstmt.executeUpdate();  
            conn.close();  
        } catch (SQLException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

What should replace // INSERT CODE HERE ?

- a. **conn.createStatement(query);**
- b. **conn.prepareStatement(query);**
- c. **conn.execute(query);**
- d. **conn.runStatement(query);**

Correct Answer:

- a. **conn.prepareStatement(query);**

Detailed Solution:

The `prepareStatement` method prepares the SQL query for execution, allowing the use of parameterized inputs.



QUESTION 6:

Which of the following SQL operations can be executed using the `executeUpdate` method in JDBC?

- I. `INSERT INTO users (id, name) VALUES (1, 'Alice');`
 - II. `UPDATE users SET name='Bob' WHERE id=1;`
 - III. `DELETE FROM users WHERE id=1;`
 - IV. `SELECT * FROM users;`
- a. I, II, and III
 - b. Only I and II
 - c. Only I
 - d. I, II, III and IV

Correct Answer:

- a. I, II, and III

Detailed Solution:

The `executeUpdate` method is used for SQL operations that modify data (`INSERT`, `UPDATE`, `DELETE`). It cannot be used for `SELECT` queries, which return a `ResultSet`.



QUESTION 7:

What is the purpose of the `DriverManager` class in JDBC?

- a. To manage database connections.
- b. To execute SQL queries.
- c. To fetch data from a database.
- d. To represent a database record.

Correct Answer:

- a. To manage database connections.

Detailed Solution:

The `DriverManager` class loads the JDBC drivers and establishes connections to databases.



QUESTION 8:

How do you establish a connection to a database using JDBC?

- a. By creating an instance of the Connection interface
- b. By using the DriverManager.getConnection() method
- c. By implementing the Connection interface
- d. By extending the Connection class

Correct Answer:

- b. By using the DriverManager.getConnection() method

Detailed Solution:

To establish a connection to a database using JDBC, you use the `DriverManager.getConnection()` method. This method takes a JDBC URL, username, and password as parameters and returns a `Connection` object, which represents a connection to the database. The JDBC URL specifies the database type, location, and other connection details.



QUESTION 9:

Which method executes a simple query and returns a single Result Set object?

- a. **executeQuery()**
- b. **executeUpdate()**
- c. **execute()**
- d. **run()**

Correct Answer:

- a. **executeQuery()**

Detailed Solution:

The executeQuery() method is used to execute a simple SQL query that returns a single ResultSet object.



QUESTION 10:

Which package in Java contains the classes and interfaces for JDBC?

- a. java.sql
- b. java.io
- c. java.db
- d. java.net

Correct Answer:

- a. java.sql

Correct Answer:

java.sql package in Java contains the classes and interfaces for JDBC.



PROGRAMMING IN JAVA

Assignment 12

TYPE OF QUESTION: MCQ

Number of questions: 10

Total marks: $10 \times 1 = 10$

QUESTION 1:

What is the correct way to create a thread in Java?

- a. Implementing the `Runnable` interface
- b. Extending the `Thread` class
- c. Both A and B
- d. None of the above

Correct Answer:

- c. Both A and B

Detailed Solution:

In Java, a thread can be created either by extending the `Thread` class or by implementing the `Runnable` interface.



QUESTION 2:

Which method starts a thread in Java?

- a. start()
- b. run()
- c. execute()
- d. init()

Correct Answer:

- a. Start()

Detailed Solution:

The `start()` method is used to begin execution of a new thread.



QUESTION 3:

Which layout arranges components in a row-based order?

- a. GridLayout
- b. FlowLayout
- c. BorderLayout
- d. CardLayout

Correct Answer:

- b. FlowLayout

Detailed Solution:

FlowLayout arranges components in a row, in the order they are added.



QUESTION 4:

What does the `setSize(400, 300)` ; method do in a JFrame?

- a. Sets the width to 300 and height to 400
- b. Sets the width to 400 and height to 300
- c. Maximizes the Jframe
- d. Closes the Jframe

Correct Answer:

- b. Sets the width to 400 and height to 300

Detailed Solution:

The `setSize(width, height)` method sets the size of the JFrame, where 400 is the width and 300 is the height.



QUESTION 5:

What does the `ResultSet` object contain in JDBC?

- a. Only non-null records from a table
- b. All records from a table, including null values
- c. Only null records
- d. A single column from a table

Correct Answer:

- b. The `ResultSet` object contains **all** records, including those with `null` values.

Detailed Solution:

The `ResultSet` object contains all records from the table, including those with null values.



QUESTION 6:

What will be the output of

```
System.out.println("hello".length());?
```

- a. 3
- b. 5
- c. 6
- d. 4

Correct Answer:

- b. 5

Detailed Solution:

The `length()` method returns the number of characters in a string. "hello" has 5 characters.



QUESTION 7:

What is the correct way to declare an array in Java?

- a. int arr = new int(5);
- b. int[] arr = new int[5];
- c. array arr = new int[5];
- d. int arr[] = new int();

Correct Answer:

- b. int[] arr = new int[5];

Detailed Solution:

In Java, an array is declared using `int[] arr = new int[5];` where 5 is the size.



QUESTION 8:

What will be printed by the following code?

```
System.out.println("Java".charAt(1));
```

- a. J
- b. a
- c. v
- d. A

Correct Answer:

- b. a

Detailed Solution:

Indexing in Java starts from 0, so `charAt(1)` returns the second character, which is a.



QUESTION 9:

What is printed when this code runs?

```
try {
    System.out.println("Try block");
} finally {
    System.out.println("Finally block");
}
```

- a. Only "Try block"
- b. Only "Finally block"
- c. "Try block" followed by "Finally block"
- d. Compilation error

Correct Answer:

- c. "Try block" followed by "Finally block"

Detailed Solution:

The `finally` block always executes, even if no exception occurs.



QUESTION 10:

Which Java GUI framework is lightweight?

- a. AWT
- b. Swing
- c. Both A and B
- d. None

Correct Answer:

- b. Swing

Detailed Solution:

Swing is a lightweight GUI framework compared to AWT, which is heavyweight.