**Assessment-1**

50 MCQ questions and 2 coding questions

**Core Java**

1. **Which of the following is not a Java feature?**
   * A. Dynamic
   * B. Architecture Neutral
   * C. Use of Pointers
   * D. Object-oriented
   * **Answer: C**
2. **Which of the following is the correct syntax for a main method in Java?**
   * A. public void main(String args[])
   * B. public static void main(String args[])
   * C. public static int main(String args[])
   * D. public int main(String args[])
   * **Answer: B**
3. **Which data type is used to create a variable that should store text?**
   * A. myString
   * B. string
   * C. String
   * D. Txt
   * **Answer: C**
4. **Which of the following is a valid declaration of a char?**
   * A. char ch = '\utea';
   * B. char ca = 'tea';
   * C. char cr = \u0223;
   * D. char cc = '\itea';
   * **Answer: C**
5. **Which method can be used to find the length of a string in Java?**
   * A. length()
   * B. getSize()
   * C. getLength()
   * D. size()
   * **Answer: A**
6. **Which of the following is a valid keyword in Java?**
   * A. interface
   * B. string
   * C. Float
   * D. unsigned
   * **Answer: A**
7. **Which of the following is not an access modifier in Java?**
   * A. protected
   * B. public
   * C. void
   * D. private
   * **Answer: C**
8. **Which of these cannot be used for a variable name in Java?**
   * A. identifier
   * B. variable
   * C. 1st\_variable
   * D. $var
   * **Answer: C**
9. **Which of the following loops will execute the body of loop even when the condition controlling the loop is initially false?**
   * A. for
   * B. while
   * C. do-while
   * D. none
   * **Answer: C**
10. **What is the size of a byte variable in Java?**
    * A. 4 bits
    * B. 8 bits
    * C. 16 bits
    * D. 32 bits
    * **Answer: B**

**Java 8 Features**

1. **Which of the following is a new feature in Java 8?**
   * A. String concatenation
   * B. Lambdas
   * C. Swing
   * D. Collections
   * **Answer: B**
2. **What is the use of the default keyword in interfaces in Java 8?**
   * A. To define the default implementation of a method
   * B. To mark the method for deletion
   * C. To overload the method
   * D. To define a new class
   * **Answer: A**
3. **Which of the following is true about functional interfaces in Java 8?**
   * A. They must contain exactly one abstract method
   * B. They can contain multiple abstract methods
   * C. They must be annotated with @FunctionalInterface
   * D. They can be used as anonymous classes only
   * **Answer: A**
4. **Which package contains the Stream API in Java 8?**
   * A. java.lang
   * B. java.io
   * C. java.util.stream
   * D. java.util.function
   * **Answer: C**
5. **Which of the following is not a valid method of java.util.stream.Stream?**
   * A. filter
   * B. map
   * C. collect
   * D. parse
   * **Answer: D**
6. **Which of the following is used to create an infinite stream in Java 8?**
   * A. Stream.generate()
   * B. Stream.infinite()
   * C. Stream.create()
   * D. Stream.repeat()
   * **Answer: B**
7. **What does the Optional class in Java 8 help to address?**
   * A. NullPointerExceptions
   * B. Compile-time errors
   * C. Infinite loops
   * D. Incorrect type casting
   * **Answer: A**
8. **Which method is used to check if an Optional contains a value in Java 8?**
   * A. hasValue()
   * B. isPresent()
   * C. exists()
   * D. valuePresent()
   * **Answer: B**
9. **What type of expression is a lambda expression in Java 8?**
   * A. Anonymous function
   * B. Named function
   * C. Parameterized function
   * D. Static function
   * **Answer: A**
10. **Which functional interface does the forEach method in Java 8 accept?**
    * A. Consumer
    * B. Supplier
    * C. Function
    * D. Predicate
    * **Answer: A**

**JDBC**

1. **Which of the following is not a JDBC driver type?**
   * A. JDBC-ODBC bridge driver
   * B. Native-API driver
   * C. Network Protocol driver
   * D. Embedded driver
   * **Answer: D**
2. **Which method is used to execute a SELECT query in JDBC?**
   * A. execute()
   * B. executeQuery()
   * C. executeUpdate()
   * D. executeSelect()
   * **Answer: B**
3. **Which method is used to execute an INSERT, UPDATE, or DELETE query in JDBC?**
   * A. execute()
   * B. executeQuery()
   * C. executeUpdate()
   * D. executeInsert()
   * **Answer: C**
4. **Which JDBC type represents a SQL DATE?**
   * A. java.sql.Date
   * B. java.sql.Timestamp
   * C. java.sql.Time
   * D. java.util.Date
   * **Answer: A**
5. **Which method is used to load a JDBC driver class in Java?**
   * A. Class.forName()
   * B. DriverManager.load()
   * C. Driver.load()
   * D. DriverManager.registerDriver()
   * **Answer: A**
6. **What does the getConnection method of DriverManager return?**
   * A. Statement
   * B. Connection
   * C. ResultSet
   * D. SQLWarning
   * **Answer: B**
7. **Which of the following is not a valid JDBC URL?**
   * A. jdbc:mysql://localhost:3306/mydb
   * B. jdbc:oracle:thin:@localhost:1521
   * C. jdbc:sqlserver://localhost;databaseName=mydb
   * D. jdbc:database://localhost/mydb
   * **Answer: D**
8. **How do you set a string parameter in a PreparedStatement in JDBC?**
   * A. setString(index, value)
   * B. setParameter(index, value)
   * C. setStringValue(index, value)
   * D. setParam(index, value)
   * **Answer: A**
9. **Which method is used to close a Connection object in JDBC?**
   * A. closeConnection()
   * B. end()
   * C. close()
   * D. disconnect()
   * **Answer: C**
10. **What exception is thrown by JDBC methods?**
    * A. SQLException
    * B. IOException
    * C. ClassNotFoundException
    * D. ConnectionException
    * **Answer: C**

**Collections**

1. **Which of the following is not a part of the Java Collections Framework?**
   * A. HashMap
   * B. Hashtable
   * C. TreeSet
   * D. Array
   * **Answer: D**
2. **Which class provides a resizable array implementation in Java?**
   * A. Array
   * B. Arrays
   * C. ArrayList
   * D. LinkedList
   * **Answer: C**
3. **Which of the following is a marker interface?**
   * A. Serializable
   * B. Cloneable
   * C. EventListener
   * D. All of the above
   * **Answer: D**
4. **Which method is used to add an element at a specific position in a list?**
   * A. add()
   * B. insert()
   * C. addAt()
   * D. add(index, element)
   * **Answer: D**
5. **Which interface does java.util.HashSet implement?**
   * A. List
   * B. Collection
   * C. Set
   * D. Map
   * **Answer: C**

**Exception Handling**

1. **Which of the following is a checked exception?**
   * A. IOException
   * B. NullPointerException
   * C. ArrayIndexOutOfBoundsException
   * D. ArithmeticException
   * **Answer: D**
2. **Which keyword is used to handle exceptions in Java?**
   * A. try
   * B. catch
   * C. throw
   * D. All of the above
   * **Answer: D**
3. **Which block is always executed in Java, whether an exception is handled or not?**
   * A. try
   * B. catch
   * C. finally
   * D. throw
   * **Answer: C**
4. **Which of the following is true about the throw keyword in Java?**
   * A. Used to declare an exception
   * B. Used to throw an exception
   * C. Used to handle an exception
   * D. Used to ignore an exception
   * **Answer: B**
5. **Which of the following is a runtime exception?**
   * A. IOException
   * B. SQLException
   * C. ClassNotFoundException
   * D. NullPointerException
   * **Answer: D**

**Multithreading**

1. **Which method must be implemented by all threads?**
   * A. run()
   * B. start()
   * C. execute()
   * D. main()
   * **Answer: A**
2. **Which of the following is a valid constructor of the Thread class?**
   * A. Thread()
   * B. Thread(Runnable r)
   * C. Thread(String name)
   * D. All of the above
   * **Answer: D**
3. **Which of the following is the correct way to start a new thread?**
   * A. thread.run()
   * B. thread.start()
   * C. thread.init()
   * D. thread.execute()
   * **Answer: B**
4. **Which method is used to pause the execution of a thread for a specific period?**
   * A. suspend()
   * B. stop()
   * C. sleep()
   * D. wait()
   * **Answer: C**
5. **Which is the correct way to implement a thread in Java?**
   * A. Extend the Thread class
   * B. Implement the Runnable interface
   * C. Both A and B
   * D. None of the above
   * **Answer: C**
6. **Which method is used to wait for a thread to finish its execution?**
   * A. wait()
   * B. notify()
   * C. join()
   * D. finish()
   * **Answer: B**
7. **What does the synchronized keyword in Java do?**
   * A. Allows multiple threads to access a resource
   * B. Prevents multiple threads from accessing a resource simultaneously
   * C. Stops a thread from executing
   * D. Starts a new thread
   * **Answer: B**
8. **Which method is called when a thread is started?**
   * A. init()
   * B. run()
   * C. start()
   * D. execute()
   * **Answer: C**
9. **Which of the following is not a state of a thread?**
   * A. New
   * B. Runnable
   * C. Blocked
   * D. Finished
   * **Answer: D**
10. **What does the wait() method do in a synchronized block?**
    * A. Releases the lock and waits until notify() or notifyAll() is called
    * B. Acquires the lock and waits
    * C. Continues execution without waiting
    * D. Throws an exception
    * **Answer: A**

**Coding question no:1**

**Scenario-Based Question: CRUD Application Using Collections in Java**

**Scenario:**

You are tasked with developing a simple CRUD application for managing a list of Student objects. Each Student has an id, name, and age. The application should allow the following functionalities:

1. **Create**: Add a new student to the list.
2. **Read**: Display all students or find a student by their id.
3. **Update**: Update the details of a student by their id.
4. **Delete**: Remove a student from the list by their id.

**Requirements:**

* Use ArrayList or HashMap to store the Student objects.
* Ensure that the Student class has appropriate constructors, getters, and setters.

**Instructions:**

Implement the following methods in a Java class named StudentManager:

1. **addStudent(Student student)**: Adds a new student to the collection.
2. **getAllStudents()**: Returns a list of all students.
3. **getStudentById(int id)**: Returns a student by their id, or null if not found.
4. **updateStudent(int id, String name, int age)**: Updates the name and age of the student with the specified id.
5. **deleteStudent(int id)**: Removes the student with the specified id from the collection.

**Coding question:2**

### **Scenario-Based Question: CRUD Application Using JDBC with Oracle**

**Scenario:**

You are tasked with developing a simple CRUD application for managing a list of Student objects stored in an Oracle database. Each Student has an id, name, and age. The application should allow the following functionalities:

1. **Create**: Add a new student to the database.
2. **Read**: Retrieve all students or find a student by their id.
3. **Update**: Update the details of a student by their id.
4. **Delete**: Remove a student from the database by their id.

**Database Table Schema:**

sql

Copy code

CREATE TABLE Students (

id NUMBER PRIMARY KEY,

name VARCHAR2(50),

age NUMBER

);

**Requirements:**

* Use JDBC to interact with the Oracle database.
* Ensure that the Student class has appropriate constructors, getters, and setters.

**Assesment-2**

### Spring Core

1. **Which of the following is not a core module of the Spring Framework?**
   * A. Spring AOP
   * B. Spring MVC
   * C. Spring ORM
   * D. Spring Data
   * **Answer: D**
2. **Which of the following is a benefit of using Spring Framework?**
   * A. Lightweight
   * B. Inversion of Control
   * C. Aspect-Oriented Programming
   * D. All of the above
   * **Answer: D**
3. **Which annotation is used to define a Spring bean?**
   * A. @Component
   * B. @Bean
   * C. @Service
   * D. All of the above
   * **Answer: D**
4. **What is the default scope of a Spring bean?**
   * A. Singleton
   * B. Prototype
   * C. Request
   * D. Session
   * **Answer: C**
5. **Which of the following is used for dependency injection in Spring?**
   * A. Constructor injection
   * B. Setter injection
   * C. Field injection
   * D. All of the above
   * **Answer: D**

### Spring JDBC

1. **Which class is used to simplify database operations in Spring JDBC?**
   * A. JdbcTemplate
   * B. HibernateTemplate
   * C. JpaTemplate
   * D. SqlTemplate
   * **Answer: A**
2. **Which method in JdbcTemplate is used to execute a query that returns a single row?**
   * A. queryForMap
   * B. queryForObject
   * C. queryForList
   * D. execute
   * **Answer: B**
3. **How do you handle exceptions in Spring JDBC?**
   * A. By using try-catch blocks
   * B. Using SQLExceptionTranslator
   * C. By implementing ErrorHandler
   * D. Using ExceptionHandler
   * **Answer:A**

### Spring MVC

1. **Which annotation is used to mark a class as a Spring MVC controller?**
   * A. @Controller
   * B. @Service
   * C. @Repository
   * D. @Component
   * **Answer: A**
2. **Which annotation is used to map a web request to a specific method in a controller?**
   * A. @RequestMapping
   * B. @GetMapping
   * C. @PostMapping
   * D. All of the above
   * **Answer: D**
3. **Which of the following is a correct way to handle exceptions in Spring MVC?**
   * A. Using @ExceptionHandler
   * B. Using @ControllerAdvice
   * C. Using HandlerExceptionResolver
   * D. All of the above
   * **Answer: D**
4. **What is the default return type of a controller method in Spring MVC?**
   * A. ModelAndView
   * B. String
   * C. void
   * D. Object
   * **Answer: B**

### Spring Security

1. **Which of the following is not a core component of Spring Security?**
   * A. Authentication
   * B. Authorization
   * C. Auditing
   * D. Encryption
   * **Answer: C**
2. **Which filter is used by Spring Security to perform authentication?**
   * A. AuthenticationFilter
   * B. UsernamePasswordAuthenticationFilter
   * C. AuthorizationFilter
   * D. LoginFilter
   * **Answer:C**
3. **How can you configure method-level security in Spring?**
   * A. Using @Secured
   * B. Using @PreAuthorize
   * C. Using @RolesAllowed
   * D. All of the above
   * **Answer: D**
4. **Which annotation is used to enable Spring Security in a Spring Boot application?**
   * A. @EnableWebSecurity
   * B. @EnableSecurity
   * C. @EnableSpringSecurity
   * D. @EnableAuthorization
   * **Answer: A**

### Hibernate

1. **Which of the following is not a Hibernate object state?**
   * A. Transient
   * B. Persistent
   * C. Detached
   * D. Static
   * **Answer: B**
2. **Which method is used to save an object in Hibernate?**
   * A. save()
   * B. persist()
   * C. saveOrUpdate()
   * D. All of the above
   * **Answer: D**
3. **Which of the following is used to configure Hibernate?**
   * A. hibernate.cfg.xml
   * B. hibernate.properties
   * C. Both A and B
   * D. None of the above
   * **Answer: C**
4. **Which annotation is used to mark a class as an entity in Hibernate?**
   * A. @Entity
   * B. @Table
   * C. @MappedEntity
   * D. @HibernateEntity
   * **Answer: A**
5. **What is the default fetching strategy in Hibernate?**
   * A. Eager
   * B. Lazy
   * C. Immediate
   * D. Preload
   * **Answer: B**

### Spring Boot

1. **Which annotation is used to enable Spring Boot auto-configuration?**
   * A. @EnableAutoConfiguration
   * B. @SpringBootApplication
   * C. @EnableConfiguration
   * D. @AutoConfiguration
   * **Answer:A**
2. **How do you run a Spring Boot application?**
   * A. Using java -jar
   * B. Using mvn spring-boot:run
   * C. Using an IDE
   * D. All of the above
   * **Answer: D**
3. **Which file is used to configure Spring Boot application settings?**
   * A. application.properties
   * B. application.yml
   * C. Both A and B
   * D. application.xml
   * **Answer: C**
4. **Which dependency is used to create a Spring Boot project?**
   * A. spring-boot-starter
   * B. spring-boot-autoconfigure
   * C. spring-boot-starter-parent
   * D. spring-boot-application
   * **Answer: A**

### Maven

1. **What is the default packaging type in Maven?**
   * A. jar
   * B. war
   * C. ear
   * D. pom
   * **Answer:D**
2. **Which file is used to configure a Maven project?**
   * A. build.xml
   * B. project.xml
   * C. pom.xml
   * D. settings.xml
   * **Answer: C**
3. **Which Maven plugin is used to compile Java code?**
   * A. maven-compiler-plugin
   * B. maven-jar-plugin
   * C. maven-war-plugin
   * D. maven-assembly-plugin
   * **Answer: A**
4. **Which of the following is not a build lifecycle in Maven?**
   * A. clean
   * B. validate
   * C. deploy
   * D. compile
   * **Answer: B**
5. **Which of the following is used to specify a dependency in Maven?**
   * A. <dependency>
   * B. <include>
   * C. <require>
   * D. <use>
   * **Answer: A**

### Git

1. **Which command is used to create a new Git repository?**
   * A. git init
   * B. git create
   * C. git new
   * D. git start
   * **Answer: A**
2. **Which command is used to clone a Git repository?**
   * A. git copy
   * B. git clone
   * C. git pull
   * D. git download
   * **Answer: B**
3. **Which command is used to stage changes for the next commit?**
   * A. git add
   * B. git commit
   * C. git push
   * D. git stage
   * **Answer: A**
4. **Which command is used to view the commit history in Git?**
   * A. git history
   * B. git log
   * C. git status
   * D. git commits
   * **Answer:B**
5. **Which command is used to merge branches in Git?**
   * A. git merge
   * B. git integrate
   * C. git join
   * D. git rebase
   * **Answer: A**

### Spring ORM with Hibernate

1. **Which class is used as the central interface for Spring's Hibernate integration?**
   * A. HibernateTemplate
   * B. HibernateManager
   * C. HibernateController
   * D. HibernateExecutor
   * **Answer: A**
2. **Which method in HibernateTemplate is used to save an entity?**
   * A. save
   * B. persist
   * C. saveOrUpdate
   * D. All of the above
   * **Answer: D**
3. **Which transaction management type is recommended in Spring applications?**
   * A. Programmatic
   * B. Declarative
   * C. Both A and B
   * D. None of the above
   * **Answer: C**
4. **Which annotation is used to enable transaction management in Spring?**
   * A. @EnableTransactionManagement
   * B. @Transaction
   * C. @Transactional
   * D. @TransactionManagement
   * **Answer: C**
5. **How do you configure a DataSource in a Spring application?**
   * A. Using Java configuration
   * B. Using XML configuration
   * C. Using application.properties or application.yml
   * D. All of the above
   * **Answer: D**

**Coding question-1:**

### Scenario-Based Question: CRUD Application Using Spring Core, Spring JDBC, and Maven

**Scenario:**

You are tasked with developing a CRUD application to manage a list of Employee objects stored in a MySQL database. Each Employee has an id, name, and department. The application should allow the following functionalities:

1. **Create**: Add a new employee to the database.
2. **Read**: Retrieve all employees or find an employee by their id.
3. **Update**: Update the details of an employee by their id.
4. **Delete**: Remove an employee from the database by their id.

**Requirements:**

* Use Spring Core for dependency injection.
* Use Spring JDBC for database operations.
* Use Maven for project management and dependencies.
* Ensure that the Employee class has appropriate constructors, getters, and setters.
* Configure the database connection using Spring.

**Database Table Schema:**

sql

Copy code

CREATE TABLE Employees (

id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(50),

department VARCHAR(50)

);

**Instructions:**

Implement the following methods in a Java class named EmployeeManager:

1. **addEmployee(Employee employee)**: Adds a new employee to the database.
2. **getAllEmployees()**: Returns a list of all employees.
3. **getEmployeeById(int id)**: Returns an employee by their id, or null if not found.
4. **updateEmployee(Employee employee)**: Updates the details of the employee.
5. **deleteEmployee(int id)**: Removes the employee with the specified id from the database.

**Coding question-2:**

### Scenario-Based Question: CRUD Application Using Spring Boot and Spring Data JPA

**Scenario:**

You are tasked with developing a CRUD application to manage a list of Employee objects stored in a MySQL database. Each Employee has an id, name, and department. The application should allow the following functionalities:

1. **Create**: Add a new employee to the database.
2. **Read**: Retrieve all employees or find an employee by their id.
3. **Update**: Update the details of an employee by their id.
4. **Delete**: Remove an employee from the database by their id.

**Requirements:**

* Use Spring Boot for rapid application development.
* Use Spring Data JPA for database operations.
* Ensure that the Employee class has appropriate annotations for JPA.
* Configure the database connection using Spring Boot properties.