Answer the following:

1. **Explain the use of Lombok annotations.**

Ans –

Use of Lombok Annotations

Lombok is a Java library that reduces boilerplate code in Java applications by automatically generating common methods like getters, setters, constructors, and toString() at compile time. It improves code readability and maintainability.

Commonly Used Lombok Annotations:

1. **@Getter and @Setter:** Automatically generates getter and setter methods for the fields of a class.

```
@Getter
@Setter
public class User {
   private String name;
   private int age;
}
```

Equivalent to manually writing:

```
public String getName() { return name; }
public void setName(String name) { this.name = name; }
```

2. **@ToString:** Generates the toString() method for the class.

```
@ToString
public class User {
   private String name;
   private int age;
}
```

3. **@NoArgsConstructor, @AllArgsConstructor, and @RequiredArgsConstructor:**
    - @NoArgsConstructor: Generates a no-argument constructor.
    - @AllArgsConstructor: Generates a constructor with all fields.
    - @RequiredArgsConstructor: Generates a constructor for final fields and fields annotated with @NonNull.

```
@NoArgsConstructor
@AllArgsConstructor
@RequiredArgsConstructor
public class User {
    private final String name;
    private int age;
}
```

4. **@Data:** Combines @Getter, @Setter, @ToString, @EqualsAndHashCode, and @RequiredArgsConstructor into a single annotation.

```
@Data
public class User {
    private String name;
    private int age;
}
```

5. **@Builder:** Enables the Builder pattern for object creation.

```
@Builder
public class User {
    private String name;
    private int age;
}
// Usage:
User user = User.builder().name("Alice").age(25).build();
```

2. What are the important methods of HTTP?

Ans-

HTTP (Hypertext Transfer Protocol) provides methods that define the operations a client can perform on a resource.

1. **GET:** Retrieves data from a server without altering the server state.
    - Example: Fetching a list of products.
    - Usage:

      GET /api/products HTTP/1.1
2. **POST:** Submits data to a server to create a new resource.
    - Example: Adding a new product.
    - Usage:

      POST /api/products HTTP/1.1

      Content-Type: application/json

      {

      "name": "Laptop",

      "price": 50000

      }
3. **PUT:** Updates an existing resource or creates a new resource if it doesn't exist.
    - Example: Updating product details.
    - Usage:

      PUT /api/products/1 HTTP/1.1

      Content-Type: application/json

      {

      "name": "Laptop",

      "price": 45000

      }

4. **DELETE:** Deletes an existing resource on the server.
    - Example: Removing a product.
    - Usage:

      DELETE /api/products/1 HTTP/1.1
5. **PATCH:** Partially updates a resource.
    - Example: Changing the price of a product.
    - Usage:

      PATCH /api/products/1 HTTP/1.1

      Content-Type: application/json

      {

      "price": 48000

      }
6. **HEAD:** Retrieves headers for a resource without the response body.
    - Example: Checking metadata like content type or length.
    - Usage:

      HEAD /api/products HTTP/1.1