**Lab 10 based Question and answers  
  
Q1. How can you ensure that a broadcast receiver does not introduce security vulnerabilities?  
Ans:**

* Register it only when needed (e.g., in code, not in the manifest).
* Use **permissions** to limit who can send broadcasts.
* Avoid using sensitive data in public broadcasts.

**Q2. What are the best practices for optimizing a service to run efficiently in the background?  
Ans:**

* Use **foreground services** for important tasks.
* Stop the service when it's done.
* Use **WorkManager** for scheduled or delayed tasks.

Q**3. How can you implement a foreground service and when should you use it instead of a background service?  
Ans:**

* Show a **notification** when starting the service.
* Use it for long or important tasks (like music or GPS tracking).
* Foreground services don’t get killed easily by the system.

**Q4. How does using LiveData or WorkManager improve the efficiency of background tasks over using a simple Service?  
Ans:**

* **LiveData** updates the UI automatically when data changes.
* **WorkManager** is better for **scheduled tasks** and **battery-friendly**.
* They handle lifecycle and system limits better than basic services.

**Q5. What are the advantages and disadvantages of binding a service to an activity?  
Ans:**

**Advantages:**

* Can communicate directly with the service.
* Get results faster.

**Disadvantages:**

* If the activity closes, the service may stop.
* More complex to manage.

**Q6. How can implicit intents be filtered to prevent unintended apps from handling sensitive operations?  
Ans:**

* Use **intent filters** and **permissions**.
* Set a specific **action**, **category**, or **data type** to limit matches

**Q7. How can you handle permission requests dynamically when using features like the camera or location services?  
Ans:**

* Check if permission is granted using ContextCompat.
* If not, request it using ActivityCompat.requestPermissions().
* Handle the result in onRequestPermissionsResult().

**Q8. Why is a BroadcastReceiver useful in an Android application?  
Ans:**

* It listens for system or custom events (like battery low or new SMS).
* Helps apps **respond to events** without always running.

**Q9. How does a background service impact battery life, and what are some best practices to optimize its performance?  
Ans:**

* Background services can drain the battery if used too long. **Best practices:**
* Stop service when done.
* Use **JobScheduler** or **WorkManager** for delayed tasks.
* Avoid unnecessary tasks in the background.

**Q10. What security concerns should be considered when using implicit intents?  
Ans:**

* Any app can respond to them.
* May leak sensitive data. **To avoid issues:**
* Validate which app handles the intent.
* Use **explicit intents** if possible.

**Q11. How can the Activity Result API improve handling results from external activities like the camera app?  
Ans:**

* It’s easier and **less error-prone** than onActivityResult.
* Cleaner and modern way to handle results (like photos).

**Q12. What are the advantages of using services for long-running tasks instead of running them in an activity?  
Ans:**

* Activities can be closed or destroyed.
* Services keep running in the background.
* Good for tasks like downloading files or playing music.

**Q13. How can you ensure that an application correctly handles broadcasted messages without conflicts with other apps?  
Ans:**

* Use **unique action names** in intents.
* Register **broadcast receivers dynamically** if possible.
* Set **priority** if needed, but use carefully.