1. Explain how the Navigator widget works in Flutter.

Answer:

- **The Navigator widget in Flutter manages a stack of pages (routes) for navigation between screens.**
- **It allows pushing a new screen onto the stack and popping the current one to go back.**
- **Each screen is represented by a Route, and Navigator provides methods like:**
    - **Navigator.push() → Navigate to a new screen.**
    - **Navigator.pop() → Go back to the previous screen.**
    - **Navigator.pushReplacement() → Replace the current screen with a new one.**

Example:

```
Navigator.push(
  context,
  MaterialPageRoute(builder: (context) => SecondScreen()),
);
Navigator.pop(context); // Go back
```

---

2. Describe the concept of named routes and their advantages over direct route navigation.

Answer:

- **Named routes are a way to define all screens with a unique string identifier in one place (usually in MaterialApp).**
- **Instead of writing navigation logic directly with MaterialPageRoute, you just refer to the route name.**

Advantages:

1. **Centralized Management → All routes are defined in one place.**
2. **Readability → Navigation uses simple names (e.g., "/home") instead of constructors.**
3. **Easier Maintenance → Changing a route only requires updating it in one place.**

4. **Supports Passing Data → Arguments can be passed easily with Navigator.pushNamed.**

Example:

```
MaterialApp(
  initialRoute: '/',
  routes: {
    '/': (context) => HomeScreen(),
    '/second': (context) => SecondScreen(),
  },
);
```

Navigate:

```
Navigator.pushNamed(context, '/second');
```

---

3. Explain how data can be passed between screens using route arguments.

Answer:

- **Flutter allows sending data to another screen while navigating.**
- **This can be done in two main ways:**

Method 1: Passing arguments with push()

```
Navigator.push(
  context,
  MaterialPageRoute(
    builder: (context) => SecondScreen(data: "Hello from Home"),
  ),
);
class SecondScreen extends StatelessWidget {
  final String data;
  const SecondScreen({required this.data});
```

```
  @override
  Widget build(BuildContext context) {
    return Text(data);
  }
}
```

Method 2: Using Named Routes with arguments

Sending data:

```
Navigator.pushNamed(
  context,
  '/second',
  arguments: "Hello from Home",
);
```

Receiving data:

```
class SecondScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    final args = ModalRoute.of(context)!.settings.arguments as String;
    return Text(args);
  }
}
```

Advantages of passing data via routes:

- **Makes apps dynamic and reusable.**
- **Keeps screens independent by passing only the required data.**