# 📝Vue.js Attendee Q&A Handout (with Answers)

## 🔗Theory & Concept Questions

**Q1: What is the Composition API?**
A: A new Vue 3 approach to organizing component logic inside `setup()`, using functions like `ref`, `reactive`, `computed`, and `watch`. It supports better logic reuse and TypeScript.

**Q2: Difference between `ref` and `reactive`?**
A: `ref` is for primitive values (number, string, boolean) and exposes `.value`. `reactive` makes entire objects or arrays reactive, tracking all nested properties.

**Q3: What is a computed property?**
A: A cached value that automatically recalculates only when its reactive dependencies change.

**Q4: What is a watcher?**
A: A function that runs custom side-effect code when reactive data changes.

**Q5: When to use computed vs watcher?**
A: Use computed to derive new reactive values for templates. Use watch when you need to run logic or side effects when data changes.

## 🛠️Code & Demo Related Questions

**Q6: Why do we have `setup()`?**
A: It's the entry point for the Composition API where we define state, computed properties, and methods before the component renders.

**Q7: Why use `v-model`?**
A: It enables two-way binding between form inputs and component data.

**Q8: What does `v-html` do?**
A: It renders raw HTML from a string, useful for formatting like highlighting hashtags.

**Q9: Why use computed for formattedContent?**
A: It updates automatically when `post.content` changes and avoids recalculating on every render.

**Q10: Why did the like button initially fail?**
A: Because of a typo (`clike` instead of `likes`), mismatched function names, and an invalid comma in the template.

**Q11: What is `emit`?**
A: It's how child components communicate events back to their parent components.

**Q12: Why use `v-for` with `:key`?**
A: To help Vue track list items efficiently when rendering or updating.

## 🩲General / Real-World Vue Questions

**Q13: Can Vue work with backend APIs?**
A: Yes, with Fetch or Axios to call REST or GraphQL APIs.

**Q14: Vue vs React vs Angular?**
A: Vue is simpler to learn and uses templates; React uses JSX; Angular is large and opinionated with built-in features.

**Q15: Is Vue good for large apps?**
A: Yes, especially using Composition API, Vue Router, and state management like Pinia.

**Q16: When to use Vuex or Pinia?**
A: When multiple components need to share or update global data.

**Q17: How to deploy a Vue app?**
A: Build using `npm run build` and deploy the `dist` folder to hosting like Netlify or Vercel.

**Q18: Does Vue support TypeScript?**
A: Yes, especially with Vue 3 and the Composition API.

**Q19: What are single-file components?**
A: `.vue` files containing template, script, and style sections in one file.

**Q20: Can we mix Composition API and Options API?**
A: Yes, though it’s better to use one style per component.

**Q21: What are lifecycle hooks?**
A: Functions like `onMounted()` that run at specific points in a component’s life cycle.

**Q22: How do components communicate?**
A: Parent to child via props; child to parent via emit; globally via shared state (Vuex/Pinia) or an event bus.

**Q23: What are slots?**
A: Placeholders to let parent components inject custom content into child components.

**Q24: Difference between `v-if` and `v-show`?**
A: `v-if` adds/removes elements from the DOM; `v-show` just toggles CSS visibility.

**Q25: Can Vue work in existing websites?**
A: Yes, you can mount Vue components to parts of static pages.

**Q26: What are common performance tips?**
A: Use `:key` in lists, prefer computed over methods in templates, and lazy-load components.

**Q27: Can Vue do SSR?**
A: Yes, with frameworks like Nuxt.js.

**Q28: How to debug reactivity?**
A: Use Vue Devtools to inspect reactive data and component structure.

**Q29: What happens if I change a prop directly?**
A: Vue will warn; props are read-only. Instead, emit an event to ask the parent to update the value.

**Q30: How to lazy-load components?**
A: Use `defineAsyncComponent(() => import('./MyComponent.vue'))`.

## 🧭Extra Practical Tips

- Use build tools (Vite, Vue CLI) for `.vue` files and bundling.
- Scoped CSS prevents style conflicts.
- Use watch for side effects; computed for derived values.
- Always add `:key` when rendering lists with `v-for`.

---

📦**Tip:** Try Vue Devtools and test reactivity by changing data in the browser.

---

🔗*Need this as PDF? Export or ask the instructor!*