Tests that specialization is working correctly:

- Dispatch
  - On methods, includes:
    - Specialization via adding a trait bound
      - Including both remote and local traits
    - Specialization via pure structure (e.g. `(T, U)` vs `(T, T)`)
    - Specialization via concrete types vs unknown types
      - In top level of the trait reference
      - Embedded within another type (`Vec<T>` vs `Vec<i32>`)
  - Specialization based on super trait relationships
  - On assoc fns
  - Ensure that impl order doesn't matter
- Item inheritance
  - Correct default cascading for methods
  - Inheritance works across impls with varying generics
    - With projections
    - With projections that involve input types
- Normalization issues
  - Non-default assoc types can be projected
    - Including non-specialized cases
    - Including specialized cases
  - Specialized Impls can happen on projections
  - Projections and aliases play well together
  - Projections involving specialization allowed in the trait ref for impls, and overlap can still be determined
    - Only works for the simple case where the most specialized impl directly provides a non-`default` associated type
- Across crates
  - For traits defined in upstream crate
  - Full method dispatch tests, drawing from upstream crate
    - Including *additional* local specializations
  - Full method dispatch tests, *without* turning on specialization in local crate
  - Test that defaults cascade correctly from upstream crates
    - Including *additional* local use of defaults