

`std::false_type`

`std::true_type`

`detail::is_specialization
_of< Primary, Primary<
Args... > >`

```
graph BT; A["detail::is_specialization_of< Primary, Primary< Args... > >"] --> B["std::false_type"]; A --> C["std::true_type"];
```

The diagram illustrates a C++ template specialization. A base template `detail::is_specialization_of< Primary, Primary< Args... > >` is shown in a grey box at the bottom. Two blue arrows point from this base template to two white boxes above it: `std::false_type` on the left and `std::true_type` on the right. This represents the base template being specialized to either `std::false_type` or `std::true_type` based on the arguments.