

演習

演習1

a) and b)

演習2

```
template <typename T>
T min(T a, T b) {
    return (a < b) ? a : b;
}

int main() {
    std::cout << "Minimum of ints: " << min<int>(3, 5) << '\n';
    std::cout << "Minimum of doubles: " << min<double>(3.5, 1.2) << '\n';

    return 0;
}
```

演習3

```
template <typename T>
void swap(T &a, T &b) {
    T temp = a;
    a = b;
    b = temp;
}

int main() {
    int x = 5;
    int y = 10;
    std::cout << "Before swap: x = " << x << ", y = " << y << '\n';
    swap(x, y);
    std::cout << "After swap: x = " << x << ", y = " << y << '\n';

    return 0;
}
```

演習4

このコードの問題点は、sum関数テンプレートが「T result = 0;」で初期化されていることです。この初期化は数値型の場合にだけ有効です。指定した型Tのデフォルト値でresultを初期化することで修正できます。

```
template <typename T>
T sum(const std::vector<T> &data) {
    T result = T();
    for (const auto &item : data) {
        result += item;
    }
    return result;
}
```

演習5

このユースケースでは関数テンプレートの使用が適しています。関数テンプレートを使用すると、データ型ごとに別々の関数を作成することなく、さまざまなデータ型で利用可能なprint関数を作成できるからです。DRY原則に沿った手法であり、コードのメンテナンスが容易になります。