Perfect Forwarding

The classical problem: A function wants to get its arguments by reference.

```
struct BigData{
  BigData(vector<int>& d): data(d){}           // Lvalue-Ref
  BigData(const vector<int>& d): data(d){}     // const Lvalue-Ref
  ...
};
```

```
struct BigData2{
  template<typename T>
  BigData2(T&& d): data(std::forward<T>(d)){} // Lvalue- and Rvalue-Ref
};
```

➡ For n parameter 2^n function overloads are necessary.

perfectForwarding.cpp

Perfect Forwarding

Perfect forwarding enables it to write function that can identically forward its arguments.

- ➡ The lvalue and rvalue properties are respected.

- `std::forward`
  - Stroustrup: " *... a heretofore unsolved problem in C++.*"
  - Mighty tools for the writer of generic libraries
  - Typical use case:
    - Factory method
    - Constructor