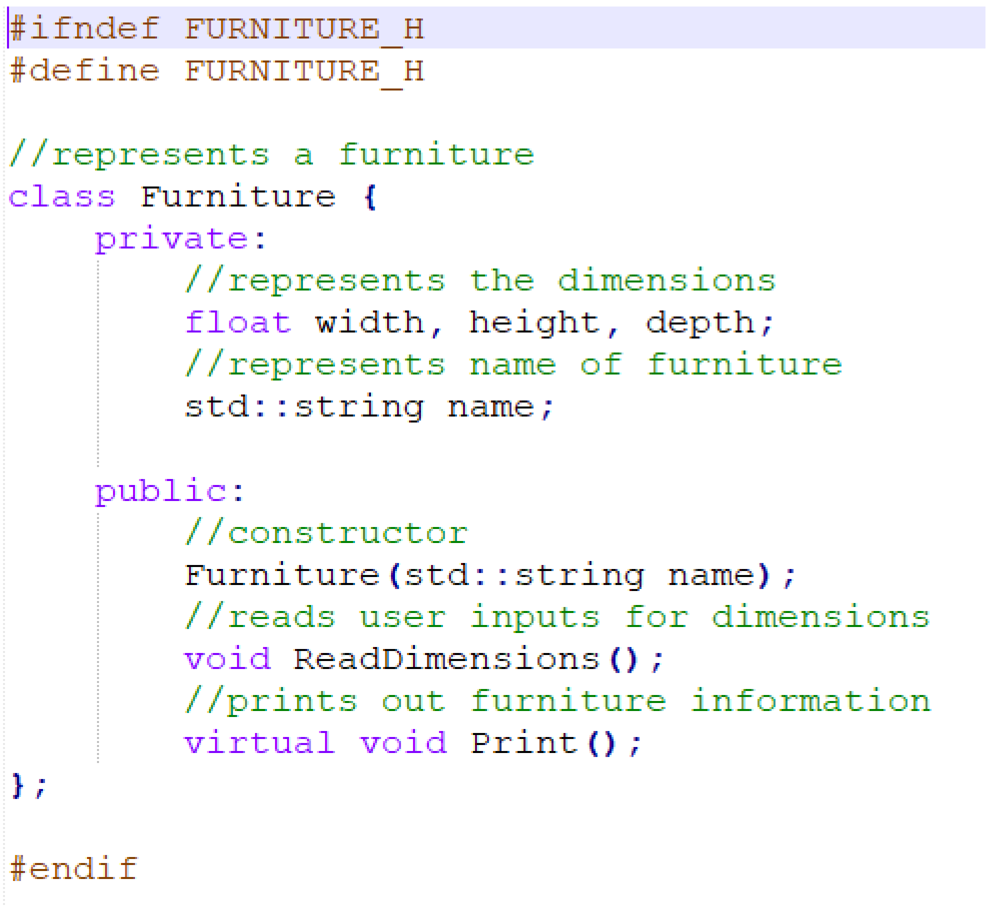
Furniture.h

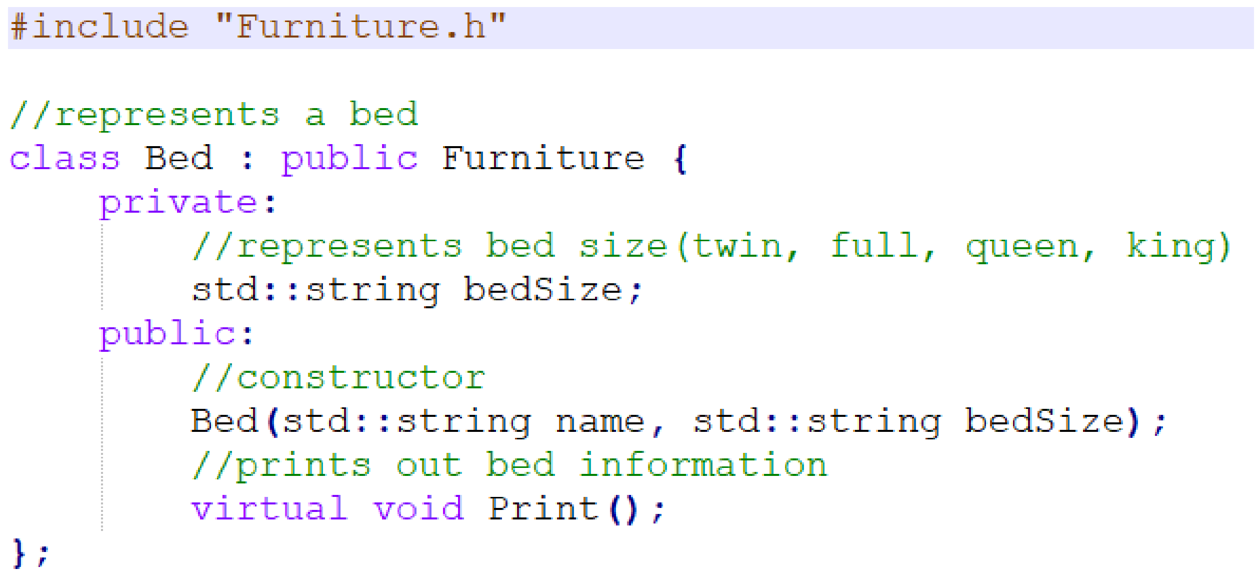


```cpp
#ifndef FURNITURE_H
#define FURNITURE_H

//represents a furniture
class Furniture {
    private:
        //represents the dimensions
        float width, height, depth;
        //represents name of furniture
        std::string name;

    public:
        //constructor
        Furniture(std::string name);
        //reads user inputs for dimensions
        void ReadDimensions();
        //prints out furniture information
        virtual void Print();
};

#endif
```

Furniture.cpp

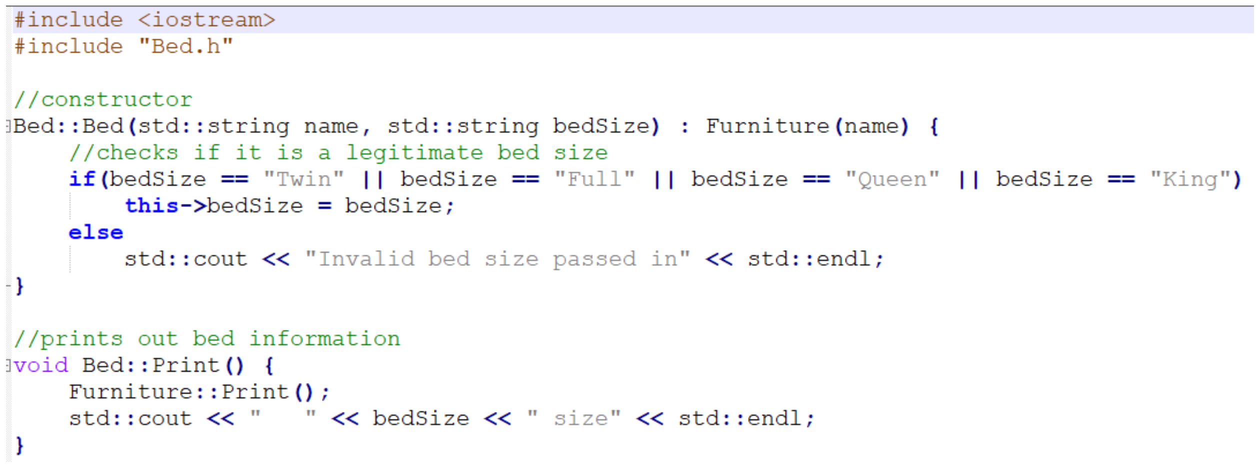
```cpp
#include <iostream>
#include "Furniture.h"

//constructor
Furniture::Furniture(std::string name) {
    this->name = name;
}

//reads user input for dimensions. Dimensions must be greater than 0
void Furniture::ReadDimensions() {
    int width = -1, height = -1, depth = -1;
    while(width < 0) {
        std::cout << "   Enter width: ";
        std::cin >> width;
        if(width < 0)
            std::cout << "Width less than 0. Please enter another number" << std::endl;
    }
    while(height < 0) {
        std::cout << "   Enter height: ";
        std::cin >> height;
        if(height < 0)
            std::cout << "height less than 0. Please enter another number" << std::endl;
    }
    while(depth < 0) {
        std::cout << "   Enter depth: ";
        std::cin >> depth;
        if(depth < 0)
            std::cout << "depth less than 0. Please enter another number" << std::endl;
    }
    this->height = height;
    this->width = width;
    this->depth = depth;
}

//prints out Furniture information such as name, width, height, depth
void Furniture::Print() {
    std::cout << name << ":" << std::endl << "   Width = " << width << ", height = " << height
    << ", depth = " << depth << std::endl;
}
```

Bed.h



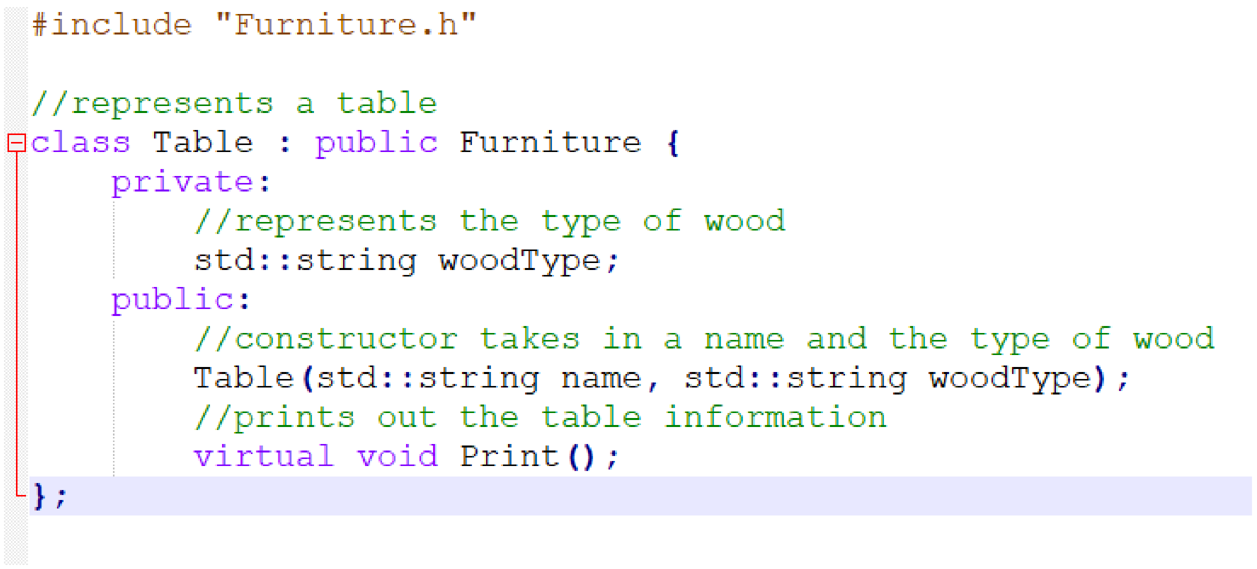
```cpp
#include "Furniture.h"

//represents a bed
class Bed : public Furniture {
    private:
        //represents bed size(twin, full, queen, king)
        std::string bedSize;
    public:
        //constructor
        Bed(std::string name, std::string bedSize);
        //prints out bed information
        virtual void Print();
};
```

Bed.cpp



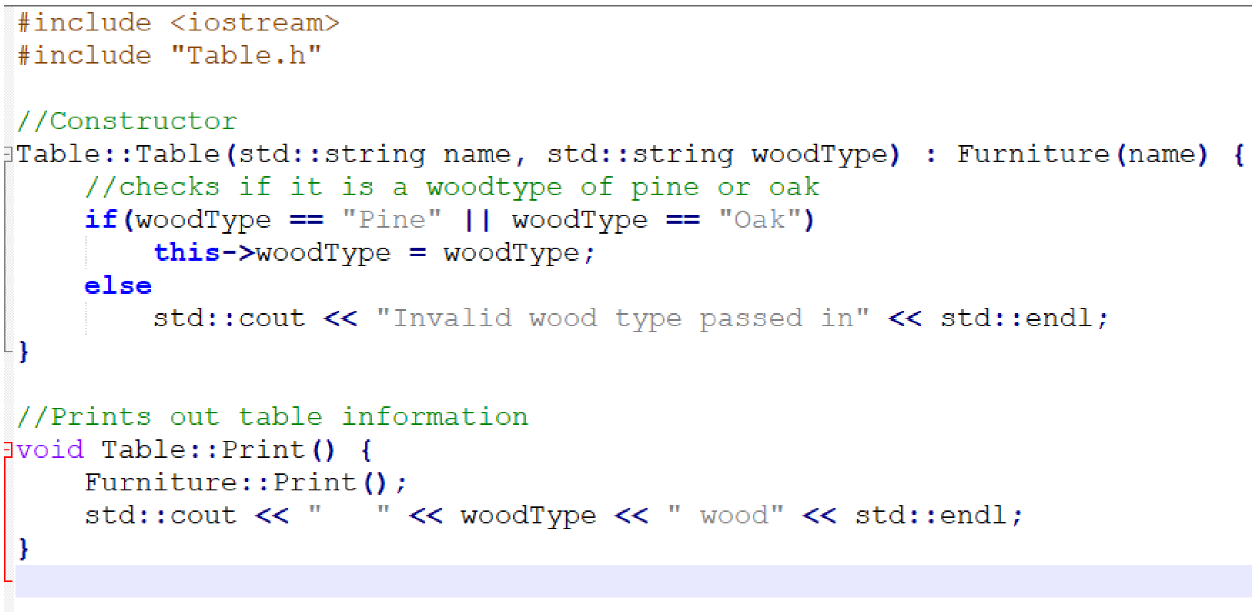
```cpp
#include <iostream>
#include "Bed.h"

//constructor
Bed::Bed(std::string name, std::string bedSize) : Furniture(name) {
    //checks if it is a legitimate bed size
    if(bedSize == "Twin" || bedSize == "Full" || bedSize == "Queen" || bedSize == "King")
        this->bedSize = bedSize;
    else
        std::cout << "Invalid bed size passed in" << std::endl;
}

//prints out bed information
void Bed::Print() {
    Furniture::Print();
    std::cout << "    " << bedSize << " size" << std::endl;
}
```

Table.h



```cpp
#include "Furniture.h"

//represents a table
class Table : public Furniture {
    private:
        //represents the type of wood
        std::string woodType;
    public:
        //constructor takes in a name and the type of wood
        Table(std::string name, std::string woodType);
        //prints out the table information
        virtual void Print();
};
```

Table.cpp



```cpp
#include <iostream>
#include "Table.h"

//Constructor
Table::Table(std::string name, std::string woodType) : Furniture(name) {
    //checks if it is a woodtype of pine or oak
    if(woodType == "Pine" || woodType == "Oak")
        this->woodType = woodType;
    else
        std::cout << "Invalid wood type passed in" << std::endl;
}

//Prints out table information
void Table::Print() {
    Furniture::Print();
    std::cout << "    " << woodType << " wood" << std::endl;
}
```

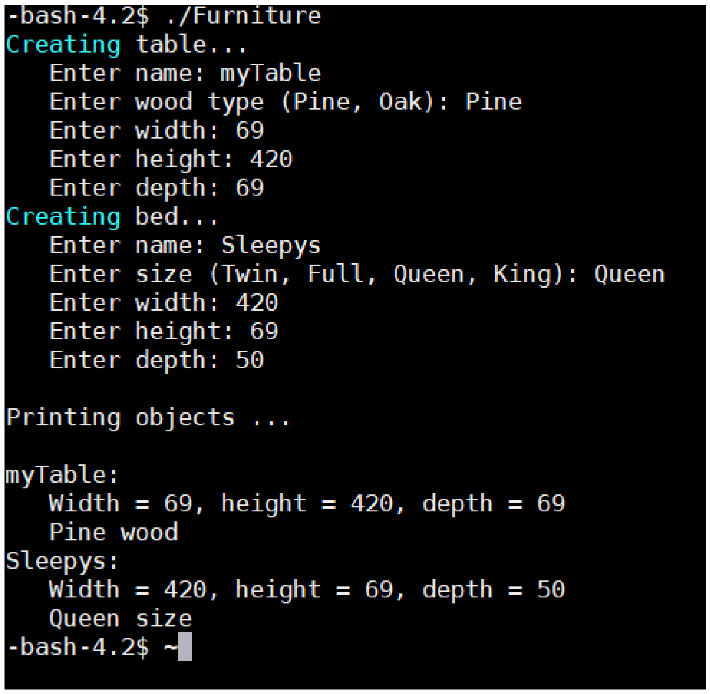
Main.cpp

```cpp
#include <iostream>
#include <string>
#include "Table.h"
#include "Bed.h"

int main() {
    std::string name, type;
    std::cout << "Creating table..." << std::endl;
    std::cout << "   Enter name: ";
    std::cin >> name;
    std::cout << "   Enter wood type (Pine, Oak): ";
    std::cin >> type;
    Table myTable(name, type);
    myTable.ReadDimensions();

    std::cout << "Creating bed..." << std::endl;
    std::cout << "   Enter name: ";
    std::cin >> name;
    std::cout << "   Enter size (Twin, Full, Queen, King): ";
    std::cin >> type;
    Bed myBed(name, type);
    myBed.ReadDimensions();

    std::cout << std::endl << "Printing objects ..." << std::endl << std::endl;
    myTable.Print();
    myBed.Print();

}
```

Makefile

```
# Furniture program Makefile

Furniture:  Furniture.o Table.o Bed.o main.o
    g++ -o Furniture Furniture.o Table.o Bed.o main.o

Furniture.o: Furniture.h Furniture.cpp
    g++ -c Furniture.cpp

Bed.o: Bed.h Bed.cpp Furniture.h
    g++ -c Bed.cpp

Table.o: Table.h Table.cpp Furniture.h
    g++ -c Table.cpp

main.o: Bed.h Furniture.h main.cpp
    g++ -c main.cpp

clean:
    rm Furniture.o Bed.o Table.o main.o Furniture
```

Console Output



```
-bash-4.2$ ./Furniture
Creating table...
   Enter name: myTable
   Enter wood type (Pine, Oak): Pine
   Enter width: 69
   Enter height: 420
   Enter depth: 69
Creating bed...
   Enter name: Sleepys
   Enter size (Twin, Full, Queen, King): Queen
   Enter width: 420
   Enter height: 69
   Enter depth: 50

Printing objects ...

myTable:
   Width = 69, height = 420, depth = 69
   Pine wood
Sleepys:
   Width = 420, height = 69, depth = 50
   Queen size
-bash-4.2$ ~
```