```
/*
 Project1
 Robert Florence
 CS 236
 C++
 Prof. Ferguson
 Classes and Objects
 Header file
 */

#ifndef __Project1__Tree__
#define __Project1__Tree__

#include <iostream>
#include <string>
#include <stdio.h>
using namespace std;

class Tree{

private:
    static float height;
    static int Numbranches;
    static string type;

public:

    Tree(); //default constructor

    Tree(int numbr, double hght, string typ); //overloaded constructor

    string getType();
    int getNumBranches();
    double getHeight();

    void setHeight(double height);
    void setNumbranch(int numbranches);
    void setType(string type);
    void toString();

#endif

};
```

```
/*
 Project1
 Robert Florence
 CS 236
 C++
 Prof. Ferguson
 Classes and Objects
  cpp file
 */

#include "Tree.h"
#include <string>
using namespace std;

Tree::Tree(){
    height = 0.0;
    Numbranches = 0;
    type = " ";

};

Tree::Tree(int number, double het, string teep){

    height = het;
    Numbranches = number;
    type = teep;

};

string Tree::getType(){
    return type;
}

int Tree::getNumBranches(){
    return Numbranches;
}

double Tree::getHeight(){
    return height;
}

void Tree::setHeight(double hght){
```

```
    if (hght > 0 && hght <= 300){
        height = hght;
    }
    else {
        cout << "Not a valid tree Height" << endl;
    }
}

void Tree::setNumbranch(int numbrnch){
    if (numbrnch > 0 && numbrnch <= 200){
        Numbranches = numbrnch;
    }
    else {
        cout << "Invalid tree Branch input" << endl;
    }

}
void Tree::setType(string typ){
    if (typ.length()>1 && typ.length()<=40){
        type = typ;
    }
    else {
        cout << "Not a valid Type of tree" << endl;
    }
}

void Tree::toString() {
    cout << "Tree Object\nHeight of Tree: " << getHeight() << endl;
    cout << "Number of branches on Tree: " << getNumBranches() << endl;
    cout << "Type of Tree: " << getType() << endl;

}
```

```
/*
 Project1
 Robert Florence
 CS 236
 C++
```

```
 Prof. Ferguson
 Classes and Objects
 Header file
 */

#ifndef __Project1__Hockey__
#define __Project1__Hockey__

#include <iostream>
#include <string>

using namespace std;
class Hockey
{
private:

    string position;
    int playernumber;
    double height;

public:

    Hockey(); // default constructor


    Hockey(int playernum, double height, string position);

    string getPosition();
    int getPlayerNumber();
    double getHeight();

    void setHeight(double height);
    void setPlayerNumber(int playernum);
    void setPosition(string position);
    void toString();

#endif
};
```

```
/*
 Project1
 Robert Florence
 CS 236
 C++
 Prof. Ferguson
 Classes and Objects
 cpp file
 */

#include "Hockey.h"
#include <string>

using namespace std;

Hockey::Hockey(){

    height = 0;
    position= " ";
    playernumber = 0;

    };

    Hockey::Hockey(int plnum, double hght, string pos){

        playernumber = plnum;
        height = hght;
        position = pos;

    };

    string Hockey::getPosition(){
        return position;
    }

    int Hockey::getPlayerNumber(){
        return playernumber;
    }

    double Hockey::getHeight(){
        return height;
    }

    void Hockey::setHeight(double Hyt){
```

```
    if (Hyt > 0 && Hyt <=110){
        height = Hyt;
    }
    else {
        cout << "Not a valid Height input, try again" << endl;
    }
}

void Hockey::setPlayerNumber(int pnu){
    if (pnu >0 && pnu <=99){
        playernumber = pnu;
    }
    else {
        cout << "Not an actual player number" << endl;
    }

}
void Hockey::setPosition(string pos){
    if (pos.length()>0 && pos.length()<=13){
        position = pos;

    }
    else {
        cout << "Invalid entry for Position" << endl;

    }
}

void Hockey::toString() {
    cout << "Hockey Object\nPosition: " << getPosition() << endl;
    cout << "Player Number: " << getPlayerNumber() << endl;
    cout << "Height: " << getHeight() << endl;

}
```

/*

```
 Project1
 Robert Florence
 CS 236
 C++
 Prof. Ferguson
 Classes and Objects
 Header file
 */

#ifndef __Project1__Date__
#define __Project1__Date__

#include <iostream>
#include <string>

using namespace std;
class Date
{
private:
    string month;
    int day;
    int year;

public:

    Date(); //default constructor


    Date(string Month, int Day, int Year); //overloaded constructor

    string getMonth();
    int getDay();
    int getYear();
    int getMonthfromNumber();

    void setDay(int day);
    void setYear(int year);
    void setMonth(string month);
    void toString();
    void printDate(int format);

#endif

};
```

```
/*
 Project1
 Robert Florence
 CS 236
 C++
 Prof. Ferguson
 Classes and Objects
  cpp file
 */

#include <boost/algorithm/string/predicate.hpp>
#include "Date.h"
#include <string>
using namespace std;


Date::Date(){
    year = 0;
    day = 0;
    month = " ";

};

Date::Date(string mon, int dy, int yr){

    day = dy;
    year = yr;
    month = mon;

};

int Date::getMonthfromNumber(){
    int mn;
    string choice;

    if (getMonth()=="1") {
        choice = "Jan";
```

```
} else if (getMonth()=="2"){
    choice = "Feb";


} else if (getMonth()=="3"){
    choice = "Mar";



} else if (getMonth()=="4"){
    choice = "Apr";



} else if (getMonth()=="5"){
    choice = "May";



} else if (getMonth()=="6"){
    choice = "Jun";



} else if (getMonth()=="7"){
    choice = "Jul";


} else if (getMonth()=="8"){
    choice = "Aug";



} else if (getMonth()=="9"){
    choice = "Sep";



} else if (getMonth()=="10"){
    choice = "Oct";



} else if (getMonth()=="11"){
    choice = "Nov";
```

```
    } else if (getMonth() == "12"){
        choice = "Dec";

    }

    return mn;
}

string Date::getMonth(){
    return month;
}

int Date::getDay(){
    return day;
}

int Date::getYear(){
    return year;
}

void Date::setDay(int dy){
    if (dy > 0 && dy <=31){
        day = dy;
    }
    else {
        cout << "Not actual Day entry, try again" << endl;
    }
}

void Date::setYear(int yr){
    if (yr >= 1970 && yr <= 2099){
        year = yr;
    }
    cout << "Year entered not within our range" << endl;

}
void Date::setMonth(string mon){
    if (mon.length()>0 && mon.length()<=11){
        month = mon;
    }
    else {
    cout << "Not a valid Month entry" << endl;
    }
}
```

```
void Date::printDate(int format){


    cout << "Format 0: ";
    cout << getMonthfromNumber()<< " " << getDay()<< ", " << getYear() << endl;

    cout << "Format 1: ";
    cout <<  << getDay()<< " " << getMonthfromNumber() << " " << getYear() << endl;

    cout << "Format 2: ";
    cout << getMonth()<< "-" << getDay()<< "-" << getYear() << endl;

    cout << "Format 3: ";
    cout << getMonth()<< "/" << getDay()<< "/" << getYear() << endl;

}


void Date::toString() {
    cout << "Date Object: " << endl;
    printDate(<#int format#>);


}
```

```
/*
Project1
Robert Florence
CS 236
 C++
```

```
 Prof. Ferguson
 Classes and Objects
Main
 */
#include "Tree.cpp"
#include "Date.cpp"
#include "Hockey.cpp"
#include <iostream>
using namespace std;


int main(int argc, const char * argv[])
{
    int choice;


    cout << "Pick which object you would like to build: " << endl;
    cout << "1 - Date, 2 - Hockey, 3 - Tree " << endl;
    cin >> choice;

    if (choice > 0 && choice < 4) {

        switch (choice) {
            case 1:
            {
                Date Date1;
                int yrob;
                int dayob;
                string monob;

                cout << "Date Object: " << endl;

                cout << "Please enter a Date (mm, dd, yyyy): " << endl;
                cin >> monob >> dayob >> yrob;

                Date1.setDay(dayob);
                 Date1.setMonth(monob);
                 Date1.setYear(yrob);

                Date1.toString();

                break;
            }

            case 2:
            {
                Hockey Hockey1;
```

```
            int plnu;
            double height;
            string position;

            cout << "Hockey Object: " << endl;

            cout << "Please enter a Position (C,LW,RW, LD, RD, G): " << endl;
            cin >> position;
            cout << "Please enter a Player Number (1-99): " << endl;
            cin >> plnu;
            cout << "Please enter a height of your player(1-110): " << endl;
            cin >> height;

            Hockey1.setHeight(height);
            Hockey1.setPlayerNumber(plnu);
            Hockey1.setPosition(position);

            Hockey1.toString();


            break;
        }

        case 3:
        {
            Tree Tree1;
            int numB;
            double hite;
            string ty;

            cout << "Tree Object: " << endl;

            cout << "Please enter a Type of Tree(oak, maple, cedar, etc): " << endl;
            cin >> ty;
            cout << "Please enter a Height of the Tree (1-300): " << endl;
            cin >> hite;
            cout << "Please enter an amount of Branches your Tree has (1-200): " <<
endl;
            cin >> numB;

            Tree1.setType(ty);
            Tree1.setHeight(hite);
            Tree1.setNumbranch(numB);

            Tree1.toString();
```

```
            break;
        }

        default:{
            cout << "No object created!" << endl;

            break;
        }
    }

} else {

    cout << "Err0r, wrong input" << endl;

}



}
```