```
//battleship class definitions
//Justin Mckenna
//12/17/2017

#include <cstdlib>
#include <ctime>


#ifndef BATTLESHIP_H_
#define BATTLESHIP_H_

const int fieldSize = 5;

// coordinates (location) of the ship and shots
class Location {
public:
       Location(); // void constructor, assigns -1 to X coord, and * to Y coord
       void pick(); // picks a random location
       void fire(); // asks the user to input coordinates of the next shot
       void print() const; // prints location in format "a1"
       friend bool compare(const Location&, const Location&);// predicate returns true if
the two locations match
       int x;  // 1 through fieldSize
       char y; // 'a' through fieldSize
};

// contains ship's coordinates (location) and whether is was sunk
class Ship {
public:
       Ship(); // void constructor, sets sunk=false
       bool match(const Location&) const; // returns true if this location matches
                                                                 // the ship's location
       bool isSunk() const { return sunk; } // checks to see if the ship is sunk
       void sink();       // sets "sunk" member variable of the ship to true
       void setLocation(const Location&); // deploys the ship at the specified location
       void printShip() const; // prints location and status of the ship
       Location loc;
private:
       bool sunk;
};

// contains the fleet of the deployed ships
class Fleet {
public:
       void showShips(int field[][fieldSize]);
       void deployFleet(); // deploys the ships in random locations
                                          // of the ocean
       bool operational() const; // predicate returns true if at least
                                                   // one ship in the fleet is not sunk
       bool isHitNSink(const Location &, int field[][fieldSize]); // returns true if
there was a deployed
                                                                 // ship at this location
(hit) and sinks it
                                                                 // otherwise returns
false (miss)
       void printFleet() const; // prints out locations of ships in fleet

private:
       static const int fleetSize = 6; // number of battleships
       int check(const Location &) const;  // returns index of the ship
                                                                     // that matches
location
                                                                     // -1 if none match
       Ship ships[fleetSize];        // battleships of the fleet
};

//deals with visual battlefield grid
void printField(int field[][fieldSize]); //prints field grid
void initialize(int field[][fieldSize]); //initializes field grid

//deals with highscores
void getScore(int numTurns);
bool bestScore(int &numTurns);
void printScore();

#endif /* BATTLESHIP_H_ */
```