```
#include <iostream>
#include <string>

struct hashable
{ virtual int hashvalue() = NULL;
  static int stringhash(string s);
  virtual void print() = NULL; };

int hashable::stringhash(string str)
{ int h=1088539;
  for (int i=0; i<str.length(); i+=1)
    h = h*69 + str[i];
  if (h<0) h=-h;
  return h; }

struct string_and_int: public hashable
{ string the_string;
  int the_int;
  string_and_int(string s, int i): the_string(s), the_int(i) { }
  virtual int hashvalue() { return stringhash(the_string); }
  virtual void print() { cout << "[" << the_string << ", " << the_int <<
"]\n"; } };

struct pussycat: public hashable
{ string name, fur_colour;
  double fluffiness;
  pussycat(string n, string fc, double fl): name(n), fur_colour(fc),
fluffiness(fl) { }
  virtual int hashvalue() { return stringhash(name); }
  virtual void print() { cout << "pussycat[" << name << ", " <<
fur_colour << ", " << fluffiness << "]\n"; } };

struct vicious_tiger: public pussycat
{ string upc;
  int number_of_people_eaten;
  vicious_tiger(string u): pussycat(u, "orange and black", 0.1), upc(u),
number_of_people_eaten(15) { }
  virtual int hashvalue() { return stringhash(upc); }
  virtual void print() { cout << "tiger[#" << upc << ", " <<
number_of_people_eaten << "\n"; } };

struct hashtable
{ static const int size = 1000;
  hashable * entry[size];
  hashtable();
  void add(hashable * item);
  hashable * find(hashable * pattern); };

hashtable::hashtable()
{ for (int i=0; i<size; i+=1)
    entry[i]=NULL; }
```

```
void hashtable::add(hashable * x)
{ int posn = abs(x->hashvalue()) % size;
  if (entry[posn]!=NULL)
  { cout << "Error! clash in table\n";
    return; }
  entry[posn]=x; }

hashable * hashtable::find(hashable * pat)
{ int posn = abs(pat->hashvalue()) % size;
  return entry[posn]; }

void main()
{ hashtable * ht = new hashtable();
  ht->add(new pussycat("tiddles", "brown", 0.7));
  ht->add(new string_and_int("seven", 7));
  ht->add(new vicious_tiger("373772832"));
  ht->add(new string_and_int("twelve", 12));
  ht->add(new pussycat("jim", "striped", 0.6));
  ht->add(new string_and_int("sixty-two", 62));

  hashable * pattern, * result;

  cout << "find jim: ";
  pattern = new pussycat("jim", "", 0);
  result = ht->find(pattern);
  if (result!=NULL)
    result->print();
  delete pattern;

  cout << "find sixty-two: ";
  pattern = new pussycat("sixty-two", "", 0);
  result = ht->find(pattern);
  if (result!=NULL)
    result->print();
  delete pattern;

  cout << "find tiddles: ";
  pattern = new pussycat("tiddles", "", 0);
  result = ht->find(pattern);
  if (result!=NULL)
    result->print();
  delete pattern; }

12:24 ~/318 $ CC univht.cpp
12:24 ~/318 $ a.out
find jim: pussycat[jim, striped, 0.6]
find sixty-two: [sixty-two, 62]
find tiddles: pussycat[tiddles, brown, 0.7]
```