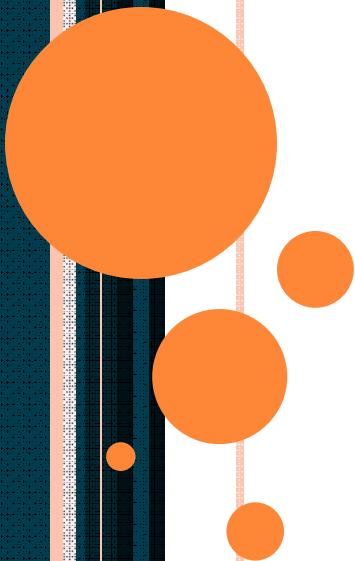# OBJECT ORIENTED PROGRAMMING USING C++

Access Functions and Friend Functions

Access functions

- To allow clients to read the value of *private* data, the class can provide a *get* function.

- To allow clients to modify *private* data, the class can provide a *set* function.

```cpp
#include <iostream.h>

class rectangle {          public:
 private:                    rectangle(float, float); // constructor
   float height;             void draw();            // draw member function
   float width;              void posn(int, int);  // position member function
   int xpos;                 void move(int, int); // move member function
   int ypos;                 float get_height();      // access function
                             void set_height(float);  // access function
                           };
```

```cpp
float rectangle::get_height()
{
 return (height);
}

void rectangle::set_height(float h)
{
 height = h;
}

void main()
{
 rectangle rc(1.0, 3.0);
 float value;

 value = rc.get_height();
 cout << "height: " << value << endl;

 rc.set_height(10.0);
 value = rc.get_height();
 cout << "height: " << value << endl;

}
```

Friend functions

- Friend functions of an object can "see" inside the object and access private member functions and data.

- To declare a function as a friend of a class, precede the function prototype in the class definition with the keyword *friend*.

```
class rectangle {

friend void set_value(rectangle&, float);   // friend declaration

private:
  float height;
  float width;
  int xpos;
  int ypos;
public:
  rectangle(float, float);   // constructor
  void draw();               // draw member function
  void posn(int, int);       // position member function
  void move(int, int);       // move member function
  float get_height();        // access function
  void set_height(float);    // access function
};
```

```cpp
void set_value(rectangle &rc, float h)
{
 rc.height = h;
}

void main()
{
 rectangle rc(1.0, 3.0);

 set_value(rc, 10.0);
}
```