Answer Key

5. Write a class, CookieJar, which has a vector of Cookie pointers. Next, write an addCookie method which takes a diameter (double), creates a Cookie on the heap, and adds it to the vector. Finally, write the Big 3 for CookieJar.

Code:

```
class CookieJar {
public:
        CookieJar() {}

        void addCookie(double diameter){
                cookies.push_back(new Cookie(diameter));
        }

        ~CookieJar(){
                for(size_t i = 0; i < cookies.size(); i++){
                        delete cookies[i];
                }
                cookies.clear();
        }

        CookieJar(const CookieJar& cj){
                for(size_t i = 0; i < cj.cookies.size(); i++){
                        Cookie * cp = new Cookie(*cj.cookies[i]);
                        cookies.push_back(cp);
                }
        }

        CookieJar& operator = (const CookieJar& cj){
                //Self-Assignment?
                if(this != &cj){
                        //Delete my data
                        for(size_t i = 0; i < cookies.size(); i++){
                                delete cookies[i];
                        }
                        cookies.clear();
                        //Copy over new data
                        for(size_t i = 0; i < cj.cookies.size(); i++){
                                Cookie * cp = new Cookie(*cj.cookies[i]);
                                cookies.push_back(cp);
                        }
                }
                return *this;
        }

private:
        vector<Cookie*> cookies;
};
```