Chapter 10 Review Questions

1. A class is user-defined type that contains data and methods (i.e. functions). Access to data and methods from outside the class can be restricted using the private `keyword` or kept `public`.

2. A class accomplishes abstraction by containing public class methods that can interact with the class object. It accomplishes encapsulation by containing the details of the class method within the class itself. It accomplishes data hiding by using private visibility for data members.

3. A class is a user-defined type. An object is a data object which uses the class definition (e.g., similar to the relationship of a variable type such as `int` to a specific variable)..

4. Variables established within an object are unique to that specific object. However, methods are shared by all class objects.

5.

```
class bankAccount
{
private:
  std::string m_name;
  std::string m_accountNum;
  double m_balance;
public:
  bankAccount(std::string name, std::string accountNum,
       double balance = 0);
  void deposit(double money);
  void withdraw(double money);
  void display();
};
```

6. Class constructors are called when an object is initialized. Class destructors are called when the object goes out of scope.

7.

```
bankAccount(std::string name, std::string accountNum,
       double balance = 0)
{
  m_name = name;
  m_accountNum = accountNum;
  m_balance = balance;
}
```

8. A default constructor is a constructor that is used to create an object when you don't provide explicit initialization values. This prevents a program from creating objects with uninitialized data members.
9.

```
class Stock
{
private:
  std::string company;
  long shares;
  double share_val;
  double total_val;
  void set_tot() { total_val = shares * share_val; }
public:
  Stock();
  Stock(const std::string & co, long n = 0, double pr = 0.0);
  ~Stock();
  void buy(long num, double price);
  void sell(long num, double price);
  void update(double price);
  void show()const;
  const Stock & topval(const Stock &s) const;
  int getShares() const { return shares; };
  double getShareVal() const { return share_val; };
  double getTotVal() const { return total_val; };
  std::string getName() const { return company; };
}
```

10. `this` is a pointer which points to the object invoking a class method. Dereferencing the pointer using `*this` returns the value stored at the memory address, not the address itself