1. Create a Register class. It should have a no-arg constructor, regular constructor, copy constructor, and operator=() [as an overload]. It should use dynamic memory (i.e. new) to create an array when the object is created named transactions. Transactions should store the amount of each transaction in the next spot in the array when the function doTransaction is called, which should pass into transaction the amount of the transaction (use a counter to ensure you are putting information into the correct spot). Each transaction should also have an IDnum which will be stored in a parallel array, such that if we were to look up information, you should have a function that returns the amount and ID number for any spot in the two arrays - call that function printSpot.

A. Write the above class with all the necessary functions

B. Show in main how you create object1 and fill in some transactions with their ID number.

C. Create object2 and assign it with your copy constructor.

D. Create object3 and then use the operator=() to give it the same data object1.

E. Create a print function for the class which prints out the amount and the ID number for each transaction. Print it the information out for object3.

2. Create a Shape class that has protected variables numberOfSides, color, and dimension. Create a subclass TwoDimensional that has a constructor that uses the base class constructor but assigns dimension to 2. Create a subclasses of TwoDimensional named Circle, Square, Triangle. Each of them should have appropriate variables and functions to output area and perimeter.

A. Create a print function in Shape named print that prints the information from the variables. Have a print function in Circle, Square and Triangle that overrides the super-classes print and prints the relevant variables (such as radius for Circle) when called.

B. Create objects of Shape, Circle and Square and show that print works appropriately.

C. Make sure to use the member-initializer notation in all the constructors

D. Use a static counter to keep track of how many times a Shape has been made. Each time the constructor for Circle, Square or Triangle occurs you should have a message displaying that it is the kth shape. When it is deallocated using the destructor you should decrement the number and output that as well. In short, when an object is created or destroyed let it give a message saying so and the number shape it is.

3. The Directory class has a list of names in it. You should use dynamic memory to create an array upon instantiation. You should create a copy constructor for the directory class and an assignment operator (i.e. operator=()). You should create a function called fillDirectory which accepts a size for the directory and the names for it both of which should be given by the user. In main, create directory1 and fill it with information. Use the assignment operator=() to set directory2 equal to directory1.

4. Create a class called Student. It should have variables name, a dynamically allocated array of grades (strings), and a RAM number assigned randomly (int between 1,000 and 9,999).

A. Create student1 in main and let the user put in 3 grades using functions from Student.

B. Create an array of 3 students and let them be instantiated to the default settings by the default constructor.

C. Create a function/hack in your Student class called Doomsday (use a bool to trigger it). If the function is called and the bool is set to 1, it should erase that objects information and set it to 0, null, etc. Show Doomsday' execution by example in main.

5. Explain what the following is and why it uses & and const: Fraction& operator=(const Fraction &);

6. What is a shallow copy versus a deep copy? Give an example of each in C++.

7. In the following code, explain the code wherever an X appears:

```
M& M::operator=(const M & d);
{

  if (this != &d)   // X
  {
    delete [] entryList; //X
    maxsize = d.maxsize;
    currentsize = d.currentsize;
    entryList = new Entry[d.maxsize];
    for (int i = 0; i < currentsize; i++)
      entryList[i] = d.entryList[i];
  }

  return *this;     // X
}
```

8. Write a friend function for the Fraction class the overloads the + operator //explain your code with comments.

9. Write a memberwise function for the Fraction class that overloads the + operator //explain your code with comments

10. Explain the difference between static memory and dynamic memory and how it is used, what the implications are for C++, and the advantages for each.

11. Explain the differences between the dot operator and the arrow (->).

12. Create a Game class that uses composition to use a mover object from the Mover class and a shaker object from the Shaker class. Design the game as simple or complex as you like, as long as you are using composition. Use member-initializer notation. Show, in main, how your game works. Have a play function that outputs to the console a series of moves that mover and shaker do when play is called. //use comments.

13. What do the keywords virtual and override do in C++?

14. Give an example of how multiple inheritance can cause problems in a program. Use a Teacher that inherits from both the Faculty and CommunityMember classes to show this. Write the code to with comments to show the problem.