Instructions. There are no special instructions for these practice problems.

1. Find the g.c.d. of the numbers 70 and 30 using function gcd. To start the algorithm, let dividend = 30 and divisor = 70. Fill in the values of the variables for the first iteration, and then write down the values for as many other iterations as you need to complete the algorithm.

  Iteration #1:
   remainder =
   dividend =
   divisor =

2. Consider the class disk declared below. A disk has a radius $r$, and is centered at some point $(x; y)$ in the plane.

   ```
   class disk {
   private:
     float rad, //the radius of the disk
     xCoord, //the x coordinate of the disk
     yCoord; //the y coordinate of the disk
   public:
     ********
   }
   ```

   (a) Write a constructor that creates an uninitialized disk, that is, a disk with no radius or center.

   (b) Write a constructor that creates a disk having radius $r$ and is centered at the point $(x; y)$.

   (c) Write a member function called area that returns the area of the receiver.

   (d) Write a member function called diam that returns the diameter of the receiver.

   (e) Write a member function called circum that returns the circumference of the receiver.

   (f) Write a member function called changeRad that changes the radius of the receiver to a value $r$ which will be passed in as a value parameter.

   (g) Write a member function called move Disk that changes the center of the receiver to position $(x; y)$. The values $x$ and $y$ will be passed in as value parameters.

   (h) Write a member function called clone Disk that returns an exact copy of the receiver.
(i) Write a member function called overlap having a disk \( d \) as a value parameter and returns true if \( d \) intersects the receiver. If \( d \) does not intersect the receiver, the function returns false. Remember that true and false are integer values, 0 for false and any nonzero integer for true. Therefore, your function should return a value of type int.

3. Use the class disk declared in problem 2 to write a complete C++ program that performs the following actions in the following order.

² Creates an empty disk \( d \).
² Changes its radius to 3.25 and its center to \((4.73; -2.11)\).
² Prints out its area, diameter and circumference.

4. Use the class fraction declared in class to write a member function called powerFraction that raises a fraction to the \( n \)th power where \( n \geq 0 \) and passes the answer back to the main program. The integer \( n \) is passed into the function as a value parameter. You may use any of the member functions that we have already written (Version 2) to implement powerFraction. The receiver should not be changed by the message powerFraction.

5. Consider the class called card declared below. A card has a value: 2,3,:,:,:,10,J,Q,K,A and a suit: Hearts, Diamonds, Clubs and Spades. (‘J’ stands for ‘Jack’, ‘Q’ for ‘Queen’, ‘K’ for ‘King’ and ‘A’ for ‘Ace’). Let the values of Jacks, Queens, Kings and Aces be 11, 12, 13 and 14 respectively. In addition, to represent suits, let Hearts = 0, Diamonds = 1, Clubs = 2 and Spades = 3.

```cpp
class card {
    private:
        int val; // the value of a card
        int suit; // the suit of a card
    public:
        *********
};
```

(a) Write a constructor that creates a card with value \( v \) and suit \( s \).
(b) Write a declaration that sets up a variable called fourOfClubs of type card and initializes this variable to the 4 of clubs.
(c) Write member functions called getValue and setSuit as follows. The function getValue returns the value of the receiver while setSuit sets the suit of the receiver to \( s \).
(d) In many card games, there is a point value assigned to every card. If the value of a card is between 2 and 10, let its point value be its value. For example, the 4 of clubs in worth 4 points. Let the Jack, Queen and King of any suit be worth 10 points, and let the Ace be worth 11 points. Write a member function called computePointValue that returns the point value of the receiver.
(e) Write a member function called `displayCard` that prints the receiver in the traditional manner, for example, ‘3 of Hearts’ or ‘Jack of Diamonds’.