

FUNDAMENTALS:

Question 1:

All three loops print out the first integers from 0 to 99. Since while loops are typically used when you do NOT know how many iterations you'll have, they are not the most appropriate loop. The for loop, which counts up from 0 to 99, is the most appropriate.

2)

Logically, there should be a positive, non-zero number of students in the class. We may use a while loop or a do-while. Since the do-while loop runs the code inside at least once, it's a better choice here.

```
int numStudents = 0;
do{
    cout << "How many students are in the class?" << endl;
    cin >> numStudents;
}while(numStudents <= 0);
```

3)

```
int n = 0;
cout << "How many stars do you want?" << endl;
cin >> n;
//I should use a for loop, since I know how many stars I want
for(int i = 0; i < n; i++)
{
    cout << "*";
}
```

WRITING PROGRAMS:

1)

I'm going to use the password "password", to simplify my life.
NOTE: in a real application, you would NEVER store a password in plain text like this :-)

```
string pass = "password";
string userPass = "";

/*
    I might use a for loop here, since I know how many iterations
    I want. I could also use a while, or a do-while. I'll
    demonstrate all three. No matter what we do, I'll need a
    boolean to determine whether the loop ended or the user
    guessed right
*/

bool isGuessed = false;

//Using a for loop, we need to add an additional loop condition
//to break out of the loop.
for(int i = 0; i < 5 && isGuessed == false; i++)
{
    cout << "Enter you password: ";
    cin >> userPass;
    if(userPass == pass)
    {
        isGuessed = true;
    }
}

//using a while loop:
int i = 0;
while(i < 5 && isGuessed == false)
{
    cout << "Enter you password: ";
    cin >> userPass;
    if(userPass == pass)
    {
        isGuessed = true;
    }
    i++; //WE HAVE TO REMEMBER THIS!
}

//using a do-while
int i = 0;
do{
    cout << "Enter you password: ";
    cin >> userPass;
    if(userPass == pass)
    {
        isGuessed = true;
    }
    i++; //WE HAVE TO REMEMBER THIS!
}while(i < 5 && isGuessed == false);
```

```

//finally, we need to output
if(isGuessed == true)
{
    cout << "Your balance is $100";
}
else
{
    cout << "Your account has been locked.";
}

```

2)

```

int n = 0;
do{
    cout << "Enter a positive integer, n: ";
    cin >> n;
}while(n < 0);

//a for loop is appropriate, since we need to check the remainder
//on a known series of numbers
bool isPrime = true; //we need to break out if we found it is
//not prime
for(int i = 2; i <= sqrt(n) && isPrime == true; i++)
{
    if(n % i == 0)
    {
        //then n is divisible by i, and is not prime
        isPrime = false;
    }
}
if(isPrime == false)
{
    cout << "Not prime!";
}
else
{
    cout << "Prime!";
}

```

3)

//we need to slightly modify the code to question 2 by wrapping it
//in another for loop.

```
for(int n = 1; n <= 100; n++)
{
    bool isPrime = true; //we need to break out if we found it is
    //not prime
    for(int i = 2; i <= sqrt(n) && isPrime == true; i++)
    {
        if(n % i == 0)
        {
            //then n is divisible by i, and is not prime
            isPrime = false;
        }
    }
    if(isPrime == true)
    {
        cout << n << "is prime!" << endl;
    } //we just want to see if the primes, so just output here
}
```

4) We need to remember some physics here. The x velocity is always constant at $v_x = 300 \cdot \cos(\text{angle_in_radians})$. The y velocity begins at $v_y = 300 \cdot \sin(\text{angle_in_radians})$, but is then updated every second by accounting for gravity, which imparts a -9.8m/s/s acceleration.

Let's assume that we begin at the origin.

```
double x = 0, y = 0;
double angle_in_radians = 45*M_PI/180;
double vx = 300*cos(angle_in_radians);
double vy = 300*sin(angle_in_radians);
double g = -9.8;

while(y >= 0)
{
    x = x + vx*1; //multiply by 1 second, the delta-t.
    vy = vy - g*1;
    y = y + vy*1;
    cout << x << " " << y << endl;
}
```

5) This problem requires us to keep track of a few variables.

```
int numFries = 10, numHummus = 10, numCake = 10;
double money = 0; //starting cash.
char userChoice = 'R';

//a while loop is a good idea here, since we don't know how many
//orders we can take
while(numFries > 0 && numHummus > 0 && numCake > 0)
{
    cout << "Enter your order! F for fries, H for hummus, C for"
        << " cake: ";
    cin >> userChoice;
    if(userChoice == 'F')
    {
        numFries = numFries - 1;
        money = money + 1 + 1*0.13;
    }
    else if(userChoice == 'H')
    {
        numHummus = numHummus - 1;
        money = money + 2 + 2*0.13;
    }
    else if(userChoice == 'C')
    {
        numCake = numCake - 1;
        money = money + 3 + 3*0.13;
    }
    else
    {
        cout << "We don't have THAT..." << endl;
    }
}

//output total money
cout << "Total money: " << money;
```