1) Explain the basic operation of a non-inverting operational amplifier (op-amp) circuit using devices such as the LM358. Why are op-amps useful? Now explain the basic operation of a simple comparator such as the LM339. Finally, explain how to use a typical op-amp, such as the LM358, and some resistors to make a comparator circuit with a wide voltage difference (this is called hysteresis) between the comparator turning on or off.

2) a) Look at the schematic of the laser sensor/control circuit we are putting on each laser tag robot. Write several paragraphs explaining precisely how each part of the circuit works and be sure your explanation touches on every component present. You must NOT simply repeat how the original article described this circuit. Notice your components may vary somewhat from the original article as I made substitutions for older, unavailable components.

b) Draw a schematic of the laser sensor/control circuit the connections that need to go to Arduino and the laser to make it fully functional. Indicate the exact Arduino pin numbers you will be using.

c) Design a simple and compact circuit of LEDs and resistors that can indicate the numbers 0-7 (or some clear code for them) and easily fit into an Arduino Metro receptacles so a spectator can easily see the number being displayed.

3) Read the laser tag contest rules closely. Point out any ambiguities or unclear rules or instructions.

4) Find and read the data sheet for the Sharp GP2Y0A02 IR sensor. Draw a simple schematic showing how it connects to Arduino power, gnd, and I/O pin(s).

Notice this sensor is very non-linear in response. Work out a conversion system to convert the analog values returned by the sensor into distance of inches or centimeters. Note the minimum and maximum distances this sensor can reliably discern.

5) Write an Arduino function that wraps your answer to problem 4 into some working (and documented) Arduino code to accurately determine the distance from the sensor to an obstacle.

6) a) Write a basic pseudo-code description of your Arduino control program for the upcoming robot laser tag contest. You cannot be so general that someone who knows how to program Arduino could not even begin to implement your code with any certainty. For example,

Get to opposite corner of maze, shooting at moving targets along the way

is useless as pseudo-code since there is nothing algorithmic about this! It says nothing about possible paths, obstacles, avoiding said obstacles, laser use or hits strategy, etc. In fact, it says essentially nothing about strategy or tactics (look these up if needed!).

On the other hand,

Follow wall on right past center obstacle wall. If hit by laser, after required 180 degree spin, back up, and shoot forward and left in response. Drive forward 2 feet, turn left and drive past/around obstacle at top speed. Stop before hitting next wall, turn right, drive close to wall, stop facing wall, and do nothing until game is over.
is a strategy to exploit the rule which says being CLOSE to your goal can promote you into the next round as runner-up or maybe even win by just being closer!

Disclaimer: I will not reveal any strategies received here to anyone else in the class.

b) Design and code a collection of function/method prototypes in Arduino which if called at appropriate times is sufficient to perform all the operations needed by any robot in the contest. PAY PARTICULAR ATTENTION TO PARAMETERS AND RETURN VALUES. You do not need to have written and debugged every single function or method but you must account for essentially all the work the robot will do. YOU MUST CHOOSE REASONABLE FUNCTION/METHOD NAMES AND ADD COMMENTS TELLING THE READER WHAT EACH CODE SEGMENT DOES. Here is a very incomplete list of functions you might consider.

// Sets up Serial connection, init variables and attach interrupt calls
void setup() {...}

// Called by hardware interrupt when laser module records a hit.
// Increments hit counter, starts immunity period until spin is completed,
// ends immunity, and resets for future interrupts.
boolean hitByLaser() {...}

// Activates laser for fixed 0.5 second. Decrement (and returns) laser shots
// remaining. Starts latency timer for 4.5 seconds.
int fireLaser() {...}

// Returns distance in centimeters from Sharp sensor reading after
// pointing sensor servo at parameter angle.
int readIRSensor(int angle) {...}

// Drive forward some distance in centimeters:
// a) as a ballistic movement if checkFloor == FALSE and return FALSE.
// b) while checking QTI floor sensors. Stop forward movement and return TRUE
// if either floor sensor indicates white line. Else return false.
// c) while checking Sharp sensor for desired distance to an obstacle
boolean driveForward(int distance, boolean checkFloor , int obstacle) {...}

//Turn or pivot robot approximate degrees CW or CCW
void turn(int degrees, int direction) {...}

// Estimate both global (x,y) location and orientation theta in arena
Pose estimatePose() {...}