Below is the description of the hardware features of a desktop PC:

- Intel® Core™ 2 Duo Processor E6300 (2MB L2 Cache, 1.86GHz, 1066)
- Genuine Windows Vista™ Home Premium
- 2 GB Dual-Channel DDR2 SDRAM (667MHz), expandable to 4 GB
- 500 GB Serial ATA Hard Drive
- DVD+RW/CD-RW Drive
- 3.5" Floppy Drive and 13-in-1 Media Reader
- Graphics card: 512MB NVIDIA GeForce 7600 GS
- Sound Blaster X-FiTM XtremeMusic with Dolby 5.1
- Video: 1 DVI, VGA and 1 S-Video (with add-in PCI-Express video card)
- 9 USB 2.0 ports and 2 IEEE 1394 (FireWire) ports
- Integrated (10/100/1000) Gigabit Ethernet
- Expansion Slots: 3 PCI Slots, 1 PCIe x1 Slot, 1 PCIe x16 (Graphics) Slots, 1 PCIe x4/x8 Slot

1) Identify the major hardware components:
   - CPU
   - Main Memory
   - Secondary Storage
   - Input Devices
   - Output Devices

2) Explain why computers have both main memory and secondary memory.

3) Is the operating system (e.g., Windows Vista) hardware or software?

4) What is the role of the operating system?
5) The Fetch-Decode-Execute cycle (i.e., Instruction/Machine Cycle) is repeated all day.
   - Fetch Instruction - read instruction pointed at by the program counter (PC) from memory into Instr. Reg. (IR)
   - Decode Instruction - figure out what kind of instruction was read
   - Execute Instruction - do some operation with the specified operands to get some result and write it

During the Fetch-Decode-Execute cycle what must happen to the PC?
6) In the following C++ program, identify the following:
   • keywords/reserved words: words that have special meaning in the C++ programming language
   • programmer-defined identifiers: symbolic names in the program that the programmer creates for variable, functions, etc.
   • operators: operators perform operations on one or more operands. An operand is usually a piece of data, like a number.
   • comments: statements in the program for the human reader which are ignored by the compiler.
   • punctuation: characters that mark the beginning or ending of a statement, or separate items in a list.

```cpp
// Program to calculate miles per gallon
#include <iostream>
using namespace std;

int main() {
    double miles, gallons, MPG;
    cout << "Enter the number of miles: ";
    cin >> miles;
    cout << "Enter the number of gallons: ";
    cin >> gallons;
    MPG = miles / gallons;
    cout << "Your mileage was " << MPG << " miles per gallon." << endl;
    return 0;
} // end main
```

7) Write a simple C++ program to average three numbers. The user interaction should look like:

```
Enter the first number: 5
Enter the second number: 6
Enter the third number: 3
The average of these numbers is 4.66667
```
8) The textbook identifies the following steps in the programming process:

1. Clearly define what the program is to do.
2. Visualize the program running on the computer.
3. Use design tools such as a hierarchy chart, flowcharts, or pseudocode to create a model of the program.
4. Check the model for logical errors.
5. Type the code, save it, and compile it.
6. Correct any errors found during compilation. Repeat Steps 5 and 6 as many times as necessary.
7. Run the program with test data for input.
8. Correct any errors found while running the program. Repeat Steps 5 through 8 as many times as necessary.
9. Validate the results of the program.

a) Why not start by writing the program ("code")?

b) Why is it important to “Run the program with test data for input”?