Question 1: What is an operating system, and what does it do?

Answer 1: The operating system (OS) is a set of software routines used to interface between the applications and the hardware. Each operating system is designed to specific manufactured hardware configuration; these are called system platforms (Davis & Rajkumar, 2004). For example, the Windows OS is a system platform that is compatible with the hardware configuration of an Intel processor.

The OS supports applications used for e-mail, instant messenger, Web browsers, word processing, spreadsheets, and more. The operating system actually consists of several internal components supporting its functions: user interfaces, file management, device management, memory management, communication interfaces, and actual processor management. The OS maintains the communication between the other application layer and hardware layer. It can also establish communication among other computers through various hardware constructs and drivers (Davis & Rajkumar, 2004).

Question 2: What are the parts of a computer?

Answer 2: A computer is a complex machine with many different parts and systems. Typical components are as follows:
**Question 3:** What is the machine cycle?

**Answer 3:** The machine cycle happens every time an OS has to delegate a process. For example, if a user wants to save a file, the machine cycle enables this to happen.

**The Machine Cycle Process**

The computer only understands mathematic calculation. It starts when the system clock sends an electronic pulse that starts the instruction control unit (ICU). The ICU decides what the machine will do next.

The ICU then calls the instruction that is needed and assigns an instruction number to that instruction. The instruction is placed in the instruction register and the arithmetic and logic unit (ALU) calculates the value directed in the instruction register. It then stores the outcome in an area called the accumulator for future use. The next cycle starts again with the clock and
instruction control unit. The machine cycle identifies that an action is about to take place and assigns the activity a sequence number. The process is as follows: fetch, hold, calculate, store, and fetch again.

The machine cycle is sometimes defined as instruction time and execution time (i-time and e-time). When the data is fetched to the instruction register, it is called i-time; when the ALU calculates the data, it is called e-time.

Question 4: What are the types of data storage and memory?

Answer 4: Data storage is secondary storage, and memory is cache or RAM.

There are two places memory is normally located. The first place is in cache (the closest repository for data to the processor), and the second is random access memory (RAM), which are special chips that store information. A computer holds storage in two places: volatile and nonvolatile storage.

Volatile Storage

Volatile storage is called memory. A computer will empty its memory when the system is powered off; therefore, there are two primary varieties of cache: L1 and L2. They are both used to store data for immediate access to the processor used in calculation and accumulation. L1 is fastest. There is also another type of memory used with some systems called read-only memory (ROM) that does not allow changes to the data stored there. This is usually used for system start up.

Nonvolatile Storage

Nonvolatile storage does not disappear when power is off; therefore, it can be retrieved from the storage device while the system is running. Nonvolatile
storage, or data storage, usually consists of hard disk drive (usually in the multi-gigabyte range) that can be written to from the memory and saved for future use and other semipermanent and permanent storage like floppy disk (not used much anymore), USB (thumb) drives, CD-ROM, DVDs, and flash drives. The OS is kept in storage where it is held until called for to run the computer.

During the fetch activity, the data called may come from the cache, RAM, or storage.

**Question 5:** What happens in a computer before the operating system (OS) can be used?

**Answer 5:** When a user turns a computer or other device with an operating system on, a process occurs. For a computer, this is called the power on self-test, also known as POST. It is initiated by a small application that is hardwired into the system supported by the basic input/output system. This system is called BIOS (basic input/output) and is a read-only memory item.

The BIOS is stored predominantly on the complementary metal-oxide semiconductor (CMOS) chip. This chip can be partially programmed to store startup settings for a computer. When the POST is initiated, the internal input/output system checks the processor, the system clock, memory, and storage. It also checks other hardware attached for connection to the hardware layer. It initiates the OS that in turn loads all the drivers (programs for the hardware components) for the attached hardware allowing the drivers to work with the OS.

**Question 6:** How are stored programs and data organized in secondary storage?
Answer 6: A file system manages the data and maintains a listing of all data and files and their physical location. This data is indexed in the directory. The directory identifies the track and sector where the data is located and points to the location on the storage device. Once the data is in memory, it is available for the user. The user can utilize a read/write search command that enables a search head to move across the location of the storage disk.

The files may be held at a specific location or pieces of the data may be connected within storage. These pieces are called pointers. At the end of a file, the pointers identify where the continuation information is connected.

Question 7: What are the differences between proprietary and open source operating systems?

Answer 7: Operating systems come in two primary forms: proprietary and open source. A proprietary OS is an OS whose source code is intellectual property of an organization, such as Microsoft Windows. The OS cannot be modified by a user. Proprietary OS requires the permission and a licensing fee paid to the publisher to use the OS source code. You are not permitted to change the actual OS of the publisher, but applications can be written to interface with the OS. The most common proprietary OSs are Windows environments. Other OSs include Novell Client and Server OS, older Apple OSs, and various versions of disk operating systems (DOS). These all have the same primary functions.
Open source allows a user to use source codes and modify them to create other OSs. An example of an open source operating system is Linux. Open source is available for users to use to write programs and other OSs. The kernel (the core of the OS) may have some restrictions applied to protect copyright. The most common open source is the UNIX/Linux family of OS.

**Question 8:** What are some ways an application can modify data files?

**Answer 8:** Data must be updated and managed often. Normally, data is built in a list consisting of files (tables) set in a specific formation with records (rows) containing unique information and cells that make up the common columns. To keep the table current, a file must be able to be modified. The processes that are applied to data elements are *append, insert, and delete*. There are several ways to do this.

**Three Methods of Data Modification**

*Insertion* is putting new data into the middle of a file, *appending* is putting data at the end of a file, and *deleting* is removing any information from a file. These methods can get complicated. Usually a data structure is established as a series of sequential data elements with an internal buffer between elements (an open space).

**Insertion**

A simple insertion occurs when a new bit of information must be included and the sequence has to make room for it. This means it must shift the current data, place the new information in its buffer, and then place an end pointer into the space made in the data structure. It must then establish a new pointer in front of the new data within the system so the data can be read and used. Stacking is another insertion method. It shifts all data, creates room in front of a data structure, and establishes a new first link in the data structure. The new link is called a stacking point.

**Deleting**

Deleting data in a data structure works differently. The data to be deleted is identified, and its front pointer is removed and creates a new one, bypassing the data element to be deleted and connecting to the subsequent element is established. After the deleted element is bypassed, it is removed from the data structure. When appending a data structure, the last data element is given a
new pointer and the data is simply added at the end of the structure.

**Question 9:** What is a device driver, and what does it do?

**Answer 9:** For the operating system to control devices (such as a mouse, USB device, and more) attached to the computer system, it needs a software program. The software program helps to communicate the commands issued from a device to the hardware. The software program is how the user can interact with the device and the computer.

When the device is needed, the OS's device management function will call a program. This program is called a *device driver*. Once the device driver has been called, a series of routine instructions are issued that activate the device and establish the connection between the software application layer and the hardware layer. The internal program then tells the hardware what to do and how to do it. The I/O process is formatted to support the hardware. The hardware is usually connected to an IRQ port—usually in a one-processor computer system—and numbered 0–15 with the driver maintained in an I/O port address. IRQ stands for interrupt request line. "Prior to plug-and-play devices, users had to set IRQ values of devices manually when adding the device, such as a modem or printer, to a system" (IRQ numbers, 2006).

**Question 10:** What are the assembler, compiler, and interpreter?

**Answer 10:** The assembler, compiler, and interpreter are all different language program processes.

Because of the fact that most programs are not written in binary (machine) language, using a program is not as easy as it seems. The computer must have a special type of support program to translate the application software to machine language. There are three types of translation programs: assembler, compiler, and interpreter. Each handles instructions differently. The instructions are in the programs and are brought into memory when the processor starts a machine cycle; it is part of how an application/program is written.

- The assembler takes the programmer's source code (written program, usually C, C++, Visual Basic, Java, Java Script, or UNIX scripting) with the series of instructions and converts it to an object module (binary) then executes the instruction. The source instruction is converted to a single machine instruction.
- The compiler takes the instruction written in algebraic form as a math function, compares the syntax and rules through a compiler program,
and then converts the formula to an object module (binary) for execution. A compiler instruction can be converted to any number of instructions.

- The interpreter takes each individual component of the source code and converts each in sequence to machine language where it is stored and then executed (Davis & Rajkumar, 2004).

The assembler, compiler, and interpreter all allow the OS to support the various programming languages.

References
