Sam Stansfield

1. What is the difference between data and information?

*Data are information. (One of my huge pet peeves is using the word "data", which is always plural, in the singular. The singular of data is datum. A "field" contains a datum.)*

2. How does out text define the term *Data Structure*?

*A data structure is an organization of information, usually in memory, for better algorithm efficiency.*

3. Give an example of a built-in data structure found in most programming languages including Java.

*The Array structure is an example of a built-in data structure in Java. In reading though, I wonder if the ArrayList class is more important for building structures.*

4. According to the text what three criteria are used to determine whether a data structure is acceptable for a particular application?

a. *Speed*

b. *Memory overhead. Memory is so inexpensive now that I wonder if memory overhead is as important as it once was. If one has gigabytes worth of free memory what difference does a few megabytes make? There was a time when finding a few kB of memory in a program made the difference between working and not working. Thankfully, those days are long past. One does have to guard against being wasteful though.*

c. *Cost (Although the first two also have a cost aspect.)*

5. According to the text what factors determine the cost of software?

*According to the text, the main factor that determines the cost of software is the number of lines of code. The cost is generally proportional to the number of lines. Reusing methods and classes from other projects reduces this cost. My guess is that encapsulation actually helps keep this linear.*

*I'm a little surprised cost doesn't go like a larger power of N than that. I would think, more complexity should make software more than proportionally costly. Not convinced that writing and testing 200 lines of code takes twice as long as writing and testing 100 lines of code. Perhaps it isn't linear for small numbers. It is an estimate after all, what do the error bars look like? So many questions. I'm not convinced.*

6. It has been estimated that a program will consist of 300,000 lines of code. If the burdened cost of a programmer's efforts is $150 per hour, determine the cost of the program. **Note:** You must show how you calculated your answer or explain how you arrived at your answer!

*Using the graph in Figure 1.5 on page 10. The middle line shows the $150 hourly rate I would estimate the total cost at about $23 million.*

7. Put the following terms in size order (smallest to largest) listing them on the three lines below: **node**, **field**, **data set**

a. *field*

b. *node*

c. *data set*

8. An application is to be written (in Java) that would allow students to find out their GPA (a *double*) and their total number of credits (an *integer*), given their student number (an *integer*).

a. Use the following picture of the node used in this application. Label the field names and data types.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | studentNumber | GPA | totalCredits |
| **Data type** | integer | double | integer |

b. Give the total node width in bytes.

*Each integer has four bytes and the double has eight, for a total of sixteen.*

c. Which field name would be the key field?

*The studentNumber field.*

9. You have been asked to write a non-void, one-parameter method to access nodes in a data set.   
  
 a. What is the type of the argument passed to the method in the key field mode?

*Whatever the type of the key field is.*

b. What is the type of the returned value for the key field mode?

*Whatever the type of the node is.*

c. What is the type of the argument passed to the method in the number mode?

*It is an integer that represents the address of the requested node.*

d. What is the type of the returned value for the number mode?

*It is the same as the type of the node.*

10. Nodes are stored in a linear list.

a. What node comes just before node 6?

*Node 5 comes just before node 6.*

b. What node comes just after node 6?

*Node 7 comes after node 6.*

11. Name the four basic operations performed on data structures (as these are named in the text), and tell what each operation does.

a. *Insert places a* *node into the data structure.*

b. *Delete removes a node from the data structure.*

c. *Fetch returns the value of a node.*

d. *Update changes the value of a node.*

12. A method signature represents the first line of code you write for a method in Java. It includes the following:

* An access specifier which is a key word such as *public* followed by other optional qualifiers such as *static* or *final*.
* The data type of the object or value to be returned or *void* if none.
* The name of the method.
* The data type and name of each parameter that is passed to the method.
* An optional *throws* statement followed by the types of exceptions that the method may throw.

The method signature does not include any code from the body of method. Examples of method signatures can be found on **page 16** of the text, but don’t just copy those or you will not get the correct answer.

The nodes in a data set are objects of type: **Listing**.

a. Give the method signatures for each of the four basic operations if they are performed in the **node number mode**.

i. *public boolean insert(Listing newNode)*

ii. *public Listing fetch(int nodeNumber)*

iii. *public boolean delete(int nodeNumber)*

iv. *public boolean update(int nodeNumber, Listing newContents)*

b. Give the method signatures for each of the four basic operations if they are performed in the **key field mode**. Assume the key field is a String.

i. *public boolean insert( Listing newNode)*

ii. *public Listing fetch(keyType targetKey)*

iii. *public boolean delete(keyType targetKey)*

iv. *public boolean update(keyType targetKey, Listing newContents)*

13. Define the terms procedural abstraction and data abstraction.

a. Procedural abstraction

*Procedural abstraction means that we don't need to know the details of a procedure (method) in order to use it. We need to know its calling conventions, the kinds of output it produces and what it does in a general sense.*

b. Data abstraction.

*Similarly, we don't need to know the details of how a data structure is implemented in order to use it. The "nuts and bolts" aren't that important to a user. We need to know the signatures of the methods for the different processes on the data.*

14. Define the term encapsulation.

*The most important thing about encapsulation is that we use compiler enforced protocols for accessing data. The idea is to restrict access to data.*

15. What is the Java *keyword* that encapsulates data inside an object?

*Instances of java classes encapsulate their data. Each instance of a class has it's own values for the "instance" variables. If they are* public *they can be accessed via "dot" notation: InstanceName.variableName. So it is the* **class** *keyword that encapsulates data. This is one reason to make instance variables* **private** *and then provide methods for accessing them.*