Patrick Roanhouse

Section CS 80 1678

Assigned Exercises for Chapter 1

You need to turn in your answers only. You do NOT need to turn in the questions.

Note: These questions were provided by the publisher for chapter 1 of our textbook - **Internet & World Wide Web How to Program, 5th edition** by Paul Deitel, Harvey Deitel, and Abbey Deitel. The exercise number does not necessarily correspond to the section in which the topic is covered.

1.6 Fill in the blanks in each of the following statements:
a) The process of instructing the computer to solve a problem is called **Programming**.
b) What type of computer language uses Englishlike abbreviations for machine-language instructions? **assembly languages**
c) The level of computer language at which it's most convenient for you to write programs quickly and easily is **high-level languages**
d) The only language that a computer directly understands is called that computer's **machine language**.
e) Web 2.0 embraces an **architecture of participation** —a design that encourages user interaction and community contributions.
f) **Collective intelligence** is the concept that a large, diverse group of people will create smart ideas.

1.7 Fill in the blanks in each of the following statements:
a) **JAVA programing language** is now used to develop large-scale enterprise applications, to enhance the functionality of web servers, to provide applications for consumer devices and for many other purposes.
b) **C programing Language** initially became widely known as the development language of the UNIX operating system.
c) The Web 2.0 company **Google** is the fastest growing company ever.
d) The **C++** programming language was developed by Bjarne Stroustrup in the early 1980s at Bell Laboratories.

1.9 Fill in the blanks in each of the following statements:
a) **IPv6** is the next-generation Internet Protocol that features built-in security and a new addressing scheme, significantly expanding the number of addresses available.
b) HTML documents normally contain **Hyperlinks**, which, when clicked, load a specified web document.
c) A **URL** contains information that directs a browser to the resource that the user wishes to access; **Web servers** make such resources available to web clients.
d) The two most common HTTP request types are **GET** and **POST**.
e) Web-based applications are multitier applications. The **bottom tier** (also called the data tier or the information tier) maintains the application's data and typically stores data in a relational database management system. The **middle tier** implements business logic, controller logic and presentation logic to control interactions between the application's clients and its data. The **Top tier**, or client tier, is the application's user interface, which gathers input and displays output.
f) **Android**, the fastest growing mobile and smartphone operating system, is based on the Linux kernel and Java.

1.11 Describe the difference between client-side programming and server-side programming.

Client side programing relies on the use of JavaScript to be used to validate user input and interactions with the browser. It is all done right there on the users side in their browser without much need for GET requests unless pulling data from web servers to edit or save. does have limitations, such as browser dependency; the browser or scripting host must support the scripting language and capabilities. Server side programing connects to a database on the server side to generate customer response for clients. The web server and scripts on the server generates custom html documents and page containing requested data on the fly

1.13 (Cloud Computing) Describe three benefits of the cloud computing model.

Cloud computing allows you to use software, hardware and information computing stored in the "cloud"—i.e., accessed on remote computers via the Internet and available on demand—rather than having it stored on your personal computer. Amazon is one of the leading providers of public cloud

comput- ing services. You can rent extra storage capacity using the Amazon Simple Storage Service (Amazon S3), or augment processing capabilities with Ama- zon's EC2 (Amazon Elastic Compute Cloud). These services, allowing you to increase or decrease resources to meet your needs at any given time, are generally more cost effective than purchasing expensive hardware to ensure that you have enough storage and processing power to meet your needs at their peak levels. Business applications (such as CRM software) are often expensive, require significant hardware to run them and knowledgeable sup- port staff to ensure that they're running properly and securely. Using cloud computing services shifts the burden of managing these applications from the business to the service provider, saving businesses money.

1.17 (Watch as an Object) You're probably wearing on your wrist one of the world's most common types of objects—a watch. Discuss how each of the following terms and concepts applies to the notion of a watch: object, attributes, behaviors, class, inheritance (consider, for example, an alarm clock), abstraction, modeling, messages, encapsulation, interface and information hiding.
Hint: See section **1.14 Object Technology** in the textbook. For more information on OOP, see the **Lecture Notes** for this Unit. Just try to understand the concepts and terms. Use your own words to answer the question.

Object / Class is reusable software components so in regards to a watch you know you need a basic template to tell time. So you have attributes of if which include weather its digital or analog. If its 24 hours or 12 hour cycles. Then you have hours, minutes, and seconds. All attributes. Now behaviors could be changing the time for hours or seconds. Moving the arms on the hour hand or minute or seconds hands. Setting an alarm Now generally speaking these features would be universal to ALL time pieces so we could pass these down to other classes for digital watch, analog watch, alarm clock and that is what we call inheritance where these attributes and behaviors would be passed down to other classes. Now each of the object in the clock program is compartmentalized in a way. They can communicate but normally not allowed to know how other objects are implements and that is hiding information so it won't leak. Imagine you have alarms set for multiple different people in this program as attributes in

objects do you want someone to see what time your wake up call is? Or be able to edit it so you miss an important meeting? That is where encapsulation comes in, its that compartmentalization as well as information hiding. And now the interface is the how we would visualize to a user these. In the way of the watch again you have it have behaviors for different types of time pieces. Modeling is what we have been saying in making a planned idea of all the attributes and behaviors and objects needed to a particular program in a visual way. You can see how the base set of attributes about time and then type of watch can then link to the behaviors to allow you to change them. Its a way to pre plan before coding. With Modeling you can keep your thoughts about you and deal with a level of Abstraction to manage the complexity of designing the software to run multiple types of time pieces like a watch or alarm clock.