CS80 Assignment 1: Ryan Hoyda

1.6

a) Programming

b) Assembly Language

c) High Level Language

d) Machine Language

e) Architecture of Participation

f) Collective Intelligence

1.7

a) Java

b) C

c) Groupon

d) C++

1.9

a) IPv6

b) Hyperlinks

c) URL, Web Servers

d) Get, Post

e) Bottom Tier, Middle Tier, Top Tier

f) Android

1.11

Server-side programming is writing code that runs on the server, using languages supported by the server (such as Java, PHP, C#; it is possible to write code that executes on the server-side in JavaScript). Client-side programming is writing code that will run on the client, and is done in languages that can be executed by the browser, such as JavaScript.

Client Side Limitations – browser dependency; browser or scripting host must support the scripting language and capabilities. Security issues that can result from placing JavaScript on the client. Client-side scripts can be viewed by using the browser’s source-viewing capability.

Server Side Scripts – generate custom responses for clients. Wider range of programmatic capabilities. Also have access to server side software that extends server functionality.

1.13

1) Can use software, hardware, and information stored on the “cloud” and accessed remotely vs having it stored on personal computer. - Cost Effective vs Purchasing additional hardware etc.

2) You can rent extra storage capacity as needed, allowing you to increase or decrease resources. Scalable and Cost Effective.

3) Ability to augment processing power to meet your needs. Scalable and Cost Effective.

1.17

Object – Watch

Attributes – Color, shape, look, type of face, band, etc. Represented as part of design in initial diagrams

Behaviors – tells time in every time zone. Has multiple alarms and the ability to shoot laser beams.

Class – Engineering schematics/blueprints of our watch that houses the inner components. Can be used many times over to generate many watches (Reuse)

Inheritance – Ability to absorb the characteristics of an existing class for the purpose of replication and or customization etc.

Abstraction – What will our watch do, and how will it do it.

Modeling – Early design of our watch used for testing and debugging.

Messages – Setting a timer on a watch to go off at a certain time, or expel laser beam.

Encapsulation – Our class, or engineering schematics for our watch encapsulates all attributes, which makes it uniquely individual – self – contained.

Interface – The part our watch that the user interacts with; watch face.

Information Hiding – The inner workings of our watch, gears and components not visible to the user. (Implementation details are hidden within the objects themselves)