1. a persons interaction with the outside world occurs through information being received and sent input and output. in an interaction with a computer the user receives information that is output by the computer, and responds by providing input to the computer the users output becomes the computers input and vice versa.

2. sensory buffers

short-term memory

3. design is defined as achieving goals within constraints and encompasses work tasks data design, architectural design, interface design and component-level design and create a design model or design specification.

4. requirements , implementation and deployment

5. a cognitive model is the designers intended mental model for the user of the system a set of ideas about how it is organized and operates.

6. analyze and design user interfaces and new user-interface technologies, created software tools and development environment to facilitate the construction of graphical user interfaces, pioneered the user of voice and video in user interfaces, hypertext links, interactive tutorials and context sensitive help systems.

7.

8. services include tasks such as accessing the internet, sending a text message, or being able to get a location basically, anything the user is trying to do.

9. the grid is a handy tool for planning out interesting moments during a drag and drop interaction. it serves as a checklist to make sure there are no holes in the interaction.

10. placeholder targeting - most explicit way to preview the effect.

11. a. evaluation should not be thought of as a single phase in the design process (still less as an activity tacked on the end of the process if time permits). ideally, evaluation should occur throughout the design life cycle, with the results of the evaluation feeding back into modifications to the design. walkthroughs require a detailed review of a sequence of actions. in the code walkthrough, the sequence represents a segment of the program code that is stepped through by the reviewers to check certain characteristics (for example, that coding style is adhered to, conventions for spelling variables versus procedure calls, and to check that system-wide invariants are not violated). in the cognitive walkthrough, the sequence of actions refers to the steps that an interface will require a user to perform in order to accomplish some known task. the evaluators then step through that action sequence to check it for potential usability

problems. usually, the main focus of the cognitive walkthrough is to establish how easy a system is to learn. more specifically, the focus is on learning through exploration. experience shows that many users prefer to learn how to use a system by exploring its functionality hands on, and not after sufficient training or examination of a users manual. so the checks that are made during the walkthrough ask questions that address this exploratory learning. to do a walkthrough (the term walkthrough from now on refers to the cognitive walkthrough, and not to any other kind of walkthrough), you need four things a specification or prototype of the system. it doesnt have to be complete, but it should be fairly detailed. details such as the location and wording for a menu can make a big difference. a description of the task the user is to perform on the system. this should be a representative task that most users will want to do. a complete, written list of the actions needed to complete the task with the proposed system. an indication of who the users are and what kind of experience and knowledge the evaluators can assume about them. given this information, the evaluators step through the action sequence (identified in item III above) to critique the system and tell a believable story about its usability. to do this, for each action, the evaluators try to answer the following four questions for each step in the action sequence. is the effect of the action the same as the users

goal at that point each user action will have a specific effect within the system. is this effect the same as what the user is trying to achieve at this point for example, if a system operation will take some time, give an indication of how long and how much is complete. match between system and the real world the system should speak the users language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. follow real world conventions, making information appear in natural and logical order. user control and freedom users often choose system functions by mistake and need a clearly marked emergency exit to leave the unwanted state without having to go through an extended dialog. support undo and redo. consistency and standards users should not have to wonder whether words, situations or actions mean the same thing in different contexts. follow platform conventions and accepted standards. error prevention make it difficult to make errors. even better than good error messages is a careful design that prevents a problem from occurring in the first place. recognition rather than recall make objects, actions and options visible. the user should not have to remember information from one part of the dialog to another. instructions for use of the system should be visible or easily retrievable whenever appropriate. flexibility and efficiency of use allow users to tailor frequent actions. accelerators unseen by the novice user may often speed up

the interaction for the expert user to such an extent that the system can cater to both inexperienced and experienced users. aesthetic and minimalist design dialogs should not contain information that is irrelevant or rarely needed. every extra unit of information in a dialog competes with the relevant units of information and diminishes their relative visibility. help users recognize, diagnose and recover from errors error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution. help and documentation few systems can be used with no instructions so it may be necessary to provide help and documentation. any such information should be easy to search, focussed on the users task, list concrete steps to be carried out, and not be too large. once each evaluator has completed their separate assessment, all of the problems are collected and the mean severity ratings calculated. the design team will then determine the ones that are the most important and will receive attention first. review based experimental psychology and human computer interaction between them possess a wealth of experimental results and empirical evidence. some of this is specific to a particular domain, but much deals with more generic issues and applies in a variety of situations. examples of such issues are the usability of different menu types, the recall of command names, and the choice of icons. a final approach to expert evaluation exploits this inheritance, using previous

results as evidence to support (or refute) aspects of the design. it is expensive to repeat experiments continually and an expert review of relevant literature can avoid evaluation through user participation ||||-|||| the need to do so. it should be noted that experimental results cannot be expected to hold arbitrarily across contexts. the reviewer must therefore select evidence carefully, noting the experimental design chosen, the population of participants used, the analyses performed and the assumptions made. design methodologies, such as design rationale also have a role to play in evaluation at the design stage. design rationale provides a framework in which design options can be evaluated. by examining the criteria that are associated with each option in the design, and the evidence that is provided to support these criteria, informed judgments can be made in the design. dialog models can also be used to evaluate dialog sequences for problems, such as unreachable states, circular dialogs and complexity. models such as state transition networks are useful for evaluating dialog designs prior to implementation.

12. a. overlays instead of going to a new page, a mini-page can be displayed in a lightweight layer over the page. overlays are really just lightweight pop ups. we use the term lightweight to make a clear distinction between it and the normal idea of a browser pop up. browser pop ups are created as a new browser window. lightweight overlays are shown within the browser page as an overlay. older style

browser pop ups are undesirable because browser pop ups display a new browser window. as a result these windows often take time and a sizeable chunk of system resources to create. a modal overlay requires the user to interact with it before she can return to the application. sometimes overlays are non-modal. example netflix site. when a dvd is added to the users shipping list (queue), a confirmation overlay is shown. while it may appear that the only way to dismiss the overlay is by clicking the close box in the upper-right corner, in reality the user can click anywhere outside the overlay (in the dimmed area) and the overlay will dismiss. staying in the flow overlays are a good way to avoid sending a user to a new page. this allows the user to stay within the context of the original page. however, since overlays are quick to display and inexpensive to produce, sometimes they can be tempting to use too freely, and in the process, may actually break the users flow. anti-pattern idiot boxes this is a clear anti-pattern that should be avoided. we call these types of overlays idiot boxes. one of the clearest examples of idiot boxes is the way certain confirmation overlays were used in yahoo! photos. inlay a common idiom is to provide additional detail about items shown on a page. hovering over a movie revealed a detail overlay calling out the back-of-the-box information. details can be shown inline as well. roost allows house photos to be viewed in-context for a real estate listing with a

detail inlay considerations one of the more difficult things to do on most real estate sites is get a view of the house in context without having to navigate from page to page. the curb appeal, inside view, and backyard are all key factors in driving interest for a house. knowing this, the team at roost wanted to make it really easy to get to the photos quickly. combining inlays and overlays roosts solution was to combine several patterns. it uses the hover reveal, a contextual tools pattern, to reveal a set of tools when the user hovers over a listing. it uses the detail inlay pattern to show a carousel of photos when the user clicks on the view photoslink. it uses a detail overlay to blow up a thumbnail when clicked on. compare this to the traditional approach, one that requires the user to navigate from the listing page to a photo page and back again. the roost team actually expended a herculean effort in setting up this convenience, as it is dealing with hundreds of mls listings with different contractual requirements for displaying real estate photos. the roost team worked out the difficulties behind the scenes to create a nice user experience. use detail inlay to provide additional information in context without hiding other information. use detail inlay to avoid the anti-pattern hover and cover. make it easy to dismiss the detail inlay.

12. b. requirements what is wanted the first stage is establishing what exactly is needed. as a precursor to this it is usually

necessary to find out what is currently happening. for example, how do people currently watch movies what sort of personal appliances do they currently use there are a number of techniques used for this in hci interviewing people, videotaping them, looking at the documents and objects that they work with, observing them directly. analysis–the results of observation and interview need to be ordered in some way to bring out key issues and communicate with later stages of design models, which are a means to capture how people carry out the various tasks that are part of their work and life. we will look at scenarios, rich stories of interaction, which can be used in conjunction with a method like task analysis or on their own to record and make vivid actual interaction. these techniques can be used both to represent the situation as it is and also the desired situation. design–well, this is all about design, but there is a central stage when you move from what you want, to how to do it. there are numerous rules, guidelines and design principles. we need to record our design choices in some way and there are various notations and methods to do this, including those used to record the existing situation iteration and prototyping humans are complex and we cannot expect to get designs right first time. we therefore need to evaluate a design to see how well it is working and where there can be improvements.. some forms of evaluation can be done using the design on paper, but it is hard to get real feedback without trying it out. most user interface design therefore involves some form of prototyping, producing early versions of systems to try out with real users. implementation and deployment finally, when we are happy with our design, we need to create it and deploy it. this will involve writing code, perhaps making hardware,

writing documentation and manuals everything that goes into a real ||||.||||| user focus |.|||||||.|||| system that can be given to others golden rule of design. part of the understanding we need is about the circumstances and context of the particular design problem. however, there are also more generic concepts to understand. the designs we produce may be different, but often the raw materials are the same. this leads us to the golden rule of design interaction design basics in the case of a physical design this is obvious. look at a chair with a steel frame and one with a wooden frame. they are very different often the steel frames are tubular or thin l or h section steel. in contrast wooden chairs have thicker solid legs. if you made a wooden chair using the design for a metal one it would break; if you made the metal one in the design for the wooden one it would be too heavy to move. for human computer interaction the obvious materials are the human and the computer. that is we must n understand computers limitations, capacities, tools, platforms n understand people psychological, social aspects, human error. e must understand the fundamental materials of humancomputer interaction in order to design areas. for example, the way you fit seats and windows into an airplanes hull affects the safety and strength of the aircraft as a whole.

13. a.

14. a.

15. b. 6 principles for designing rich web experiences make it right allow input wherever you have output shorten the length of interaction make objects directly actionable keep it lightweight

stay on the page offer an invitation show transitions react immediately good example of inline editings are discoverability, complex editing and blending modes. group editing is based on symmetry of interaction and discoverability us readability. overlays dialog overlay detail overlay input overlay considerations the apple technique signifies that we have entered a special editing mode. when the icons become wiggly, it is not a large intuitive leap that the icons have become loose and thus we can rearrange them. discoverability admittedly, the feature is not very discoverable. but it can be argued that it is straightforward once discovered. however, pressing the home button deactivates the rearranging mode. this really should operate more like a toggle. a better way to exit the wiggly mode would be to press and hold down on a wiggly icon. it follows the idea that you are pushing the icon back into its fixed place. since deactivation is not the same mechanism as activation, it is a little hard to figure out how to go back into the normal display mode. visual noise putting edit links on each module can be visually noisy. an alternative approach is to use the group edit pattern to place an edit link at the page level that turns on edit links for each module. when the done editing link is clicked, the links for each module are hidden. again the trade-off is between visual noise and discoverability. the events there are at least 1HH events available for cueing the user during a drag and drop interaction page load before any interaction occurs, you can pre-signify the availability of drag and drop. for example, you could display a tip on the page to indicate draggability. mouse hover the mouse pointer hovers over an object that is draggable. mouse down the user holds down the mouse button on the draggable object. drag initiated after the mouse drag

starts (usually some threshold III pixels). drag leaues original location after the drag object is pulled from its location or object that contains it. drag re-enters original location uhen the object re-enters the original location. drag enters ualid target dragging ouer a ualid drop target. drag exits ualid target dragging back out of a ualid drop target. drag enters specific inualid target dragging ouer an inualid drop target. drag is ouer no specific target dragging ouer neither a ualid or inualid target. do you treat all areas outside of ualid targets as inualid drag hoaers ouer ualid target user pauses ouer the ualid target uithout dropping the object. this is usually uhen a spring loaded drop target can open up. for example, drag ouer a folder and pause, the folder opens reuealing a neuu area to drag into. placeholder target this is essentially the same placeholder target approach uue discussed earlier for dragging and dropping modules. the difference is that uhen moning an item in a list, uue are constrained to a single dimension. less feedback is needed. instead of a ripped-out area a simple hole can be exposed uhere the object uill be placed uhen dropped. a good example from the desktop uorld is apples iphoto. in a slideshou, you can easily rearrange the order of photos uith drag and drop. dragging the photo left or right causes the other photos to shuffle open a drop spot. collected selection and actions uhen yahoo! photos uas uorking its uay through an early design of its photo gallery the plan uas to shou all photos in a single, continuous scrolling page in a long uirtual list. the selection model is simple. photos are shoun in a single page and selection is easily understood in the context of this single page. houeuer, due to performance issues, the design uas changed. instead of a uirtual page, photos had to be chunked into pages. in order

to support collected selection, yahoo! photos introduced the concept of a tray into the interface. on any page, photos can be dragged into the tray. the tray keeps its contents as the user moves from page to page. so, adding a photo from page one and three more from page four would yield four items in the tray. as a nice touch, the tray would make itself visible (by sliding into view) even when the user was scrolled down below the fold.

16. a.