

## **Bachelor of Science in Information Technology**

### **Internet of Things (IoT)**

**CBCS (NOV-2019)**

**Q.P Code: 75425**

---

#### **Q1 a) Define and Explain Ubiquitous Computing (ubicom). (5)**

- Ubiquitous computing (or "ubicom") is a concept in software engineering and computer science where computing is made to appear anytime and everywhere. This paradigm is also described as pervasive computing, ambient intelligence or "everywhere".
- In contrast to desktop computing, ubiquitous computing can occur using any device, in any location, and in any format.
- A user interacts with the computer, which can exist in many different forms, including laptop computers, tablets and terminals in everyday objects such as a refrigerator or a pair of glasses.
- The underlying technologies to support ubiquitous computing include Internet, advanced middleware, operating system, mobile code, sensors, microprocessors, new I/O and user interfaces, networks, mobile protocols, location and positioning, and new materials.
- Each term emphasizes slightly different aspects. When primarily concerning the objects involved, it is also known as physical computing, the Internet of Things, haptic computing and "things that think".
- Ubiquitous computing touches on a wide range of research topics, including distributed computing, mobile computing, location computing, mobile networking, context-aware computing, sensor networks, human-computer interaction, and artificial intelligence.
- Application of ubicomp in IOT: (IoT) infographic includes the smart alarm clock and coffee pot connected to the work scheduler that is alerted to let the user sleep an extra 30 minutes even though there is an accident on the driving route to work, the car knows it needs gas and the train is off its schedule.

**Q1 b) List and Explain the roles of people making IoT.**

**(5)**

- People who make Internet of Things are:
  - Artist
  - Designer
  - Craftsperson
  - Hacker
  - Engineer
  - Architect
- People making IoT do not work individually for devices but work together in harmony and mutual understanding to do the task.
- **Artist** works at the forefront of technology innovation and can bring a fresh point of view on technology. They observe how people respond to these experiments in advance. Technologies can provide insights about what people need and will accept, and this can feed into the design and development of those technologies.
- **Engineering challenges** for IoT are developing and managing things, Network and Data/Cloud. Engineers must consider and include the comprehensive characterization of the environment in their designs. Engineers need to create virtual prototypes and simulate them to make sure they will work in the real world.
- IoT is all about a new experience, an experience that doesn't exist today. But an experience if designed well can change the way we live tomorrow. That's why designers are going to play a pivotal role in shaping up the future of IoT. For that the designers will also have to change, adapt and broaden their horizon about the way they think about design.
- **Artists** collaborate with designers on the time of installation of IoT devices or they can collaborate with traditional craftsperson for printmaking while thinking about the idea of IoT device.
- **Designers** and **Engineers** work together and for making the product. In this, the Designer will design the structure of a device by using the idea of Artist and Engineer will implement this design into actual.

- **“Hackers”** by their nature, have various technical and artistic interests and skills. They will include new features into the device or will test the device properly so that the device remains safe.
  - The **IoT Architect** leads the way through “the vision, strategy, architecture and shepherding of IoT solutions from inception to deployment.” Becoming an IoT Architect requires several strategies:
    - Data management practices
    - Develop Communication Skills
    - Leadership Skills
- 

**Q1 c) What is manufactured normalcy field? Explain. (5)**

- Venkatesh Rao a technology blogger came up with a term to explain how new technology is adopted. He proposed that we don't see the present the world that we live is now, as something that is changing. If we step back for a second, we do know that it has changed, although the big advances sneak up on us over time, hidden in plain sight. This concept is called manufactured normalcy field.
- For a technology to be adopted, it has to make its way inside the manufactured normalcy field. The successful user-experience the designer is one who gives user experience which does not stretch the boundaries of their particular normalcy field too far, even if the underlying technology is a huge leap ahead of the norm.
- For example, the mobile phone was first introduced as a phone that wasn't tethered to a particular location. Now broadly the same technology is used to provide a portable Internet terminal, which can play movies, carry your entire music collection, and make phone calls.
- As we know, Arthur C. Clarke has claimed that “any sufficiently advanced technology is indistinguishable from magic”. The Internet of things provides ordinary objects with some additional semi-hidden capabilities. The enchanted

objects of magic and fairy the tale is a good metaphor to help people understand IoT objects.

- Examples:
  - Where Dial takes its lead from the clock in Harry Potter which tracked the location of the members of the Weasley family. The Where Dial, by comparison, has to rely on mere technology for its capabilities; however, with the GPS chipsets in smartphones and location check-in services like Foursquare, it isn't much of a leap to also own an ornament which updates to show when you are at work or travelling, or at a restaurant.

---

**Q1 d) “Data available through IOT device belongs to public or company which implement the IOT device”. Discuss. (5)**

- There is an increase in the number of sensors being deployed, so it isn't always clear whose data is being gathered.
- For example in case of a camera deployed in an advertising hoarding which can check to see whether people are looking at the different adverts
- Now here arises the question whether the data belong to the company that installed the camera or to the members of the public who are looking at the adverts.
- So Adam Greenfield, a leading practitioner of urban computing, makes a convincing argument that in a public space this data is being generated by the public, so they should at least have equal rights to be aware of, and also have access to, that data.
- On private property, you can more easily claim that the members of the public don't have such a right, but perhaps the property owner might assert rights to the data rather than whoever installed the camera.
- And there are many places such as shopping malls which, to all intents and purposes, look and feel like public spaces, despite being privately owned.
- The participants at the Open Internet of Things Assembly coined the term data subjects to those people to whom the data

pertains, regardless of whether they owned the sensors used to gather the data or the property where the sensors were sited.

---

**Q1 e) What is DNS? How it works? (5)**

- The Domain Name System (DNS) is the phonebook of the internet. Humans access information online through domain names, like nytimes.com or espn.com. Web browsers interact through protocol (IP) addresses. DNS translates domain names to IP addresses so browsers can load Internet resources
- Each device connected to the Internet has a unique IP address which other machines use to find the device. DNS servers eliminate the need for humans to memorize IP addresses such as 192.168.1.1(in IPv4).

**How does DNS work?**

- The process of DNS resolution involves converting a hostname (such as www.example.com) into a computer-friendly IP address (such as 192.168.1.1). An IP address is given to each device on the Internet, and that address is necessary to find the appropriate Internet device - like a street address is used to find a particular home. When a user wants to load a webpage, a translation must occur between what a user types into their web browser (example.com) and the machine-friendly address necessary to locate the example.com webpage.
- In order to understand the process behind the DNS resolution, it's important to learn about the different hardware components a DNS query must pass between. For the web browser, the DNS lookup occurs "behind the scenes" and requires no interaction from the user's computer apart from the initial request.
- There are 4 DNS servers involved in loading a webpage:
  1. **DNS precursor** - The precursor can be thought of as a librarian who is asked to go find a particular book somewhere in a library. The DNS precursor is a server designed to receive queries from client machines through applications such as web browsers. Typically the

precursor is then responsible for making additional requests in order to satisfy the client's DNS query.

2. **Root name server** - The root server is the first step in translating (resolving) human-readable hostnames into IP addresses. It can be thought of like an index in a library that points to different racks of books - typically it serves as a reference to other more specific locations.
3. **TLD name server** - The top-level domain server (TLD) can be thought of as a specific rack of books in a library. This name server is the next step in the search for a specific IP address, and it hosts the last portion of a hostname (In example.com, the TLD server is "com").
4. **Authoritative name server** - This final name server can be thought of as a dictionary on a rack of books, in which a specific name can be translated into its definition. The authoritative name server is the last stop in the name server query. If the authoritative name server has access to the requested record, it will return the IP address for the requested hostname back to the DNS Re-cursor (the librarian) that made the initial request.

---

**Q1 f) What are TCP n UDP ports? Explain with examples. (5)**

**TCP**

- Stands for Transmission Control Protocol. Using this method, the computer sending the data connects directly to the computer receiving the data and stays connected for the duration of the transfer. With this method, the two computers can guarantee that the data has arrived safely and correctly, and then they disconnect the connection.
- This method of transferring data tends to be quicker and more reliable but puts a higher load on the computer as it has to monitor the connection and the data going across it.
- A real-life comparison to this method would be to pick up the phone and call a friend. You are having a conversation and when it is over, you both hang up, releasing the connection.

## **UDP**

- Stands for User Datagram Protocol. Using this method, the computer sending the data packages the information into a nice little package and releases it into the network with the hopes that it will get to the right place.
- What this means is that UDP does not connect directly to the receiving computer like TCP does, but rather sends the data out and relies on the devices in between the sending and the receiving computer to get the data where it is supposed to go properly.
- This method of transmission does not provide any guarantee that the data you send will ever reach its destination. On the other hand, this method of transmission has very low overhead and is therefore very popular to use for services that are not that important to work on the first try.
- For comparison you can compare it with the plain old US Postal Service. You place your mail in the mailbox and hope the Postal Service will get it to the proper location. Most of the time they do, but sometimes it gets lost along the way.

## **TCP and UDP ports**

- IP address is used to recognize your particular computer out of the millions of other computers connected to the Internet. When information is sent over the Internet to your computer how does your computer accept that information? It accepts that information by using TCP or UDP ports.
- An easy way to understand ports is to imagine your IP address is a cable box and the ports are the different channels on that cable box. The cable company knows how to send a cable to your cable box based upon a unique serial number associated with that box (IP Address), and then you receive the individual shows on different channels (Ports).
- Ports work the same way. You have an IP address, and then many ports on that IP address. You can have a total

of 65,535 TCP Ports and another 65,535 UDP ports. When a program on your computer sends or receives data over the Internet it sends that data to an IP address and a specific port on the remote computer and receives the data on a usually random port on its own computer. If it uses the TCP protocol to send and receive the data then it will connect and bind itself to a TCP port. If it uses the UDP protocol to send and receive data, it will use a UDP port. Note that once an application binds itself to a particular port, that port cannot be used by any other application. It is first-come, first-served.

- If you are the remote user connecting to a web server it would work in reverse. Your web browser would pick a random TCP port from a certain range of port numbers, and attempt to connect to port 80 on the IP address of the webserver. When the connection is established, the web browser will send the request for a particular web page and receive it from the webserver. Then both computers will disconnect the connection.
- A port is a 16-bit number used to identify specific applications and services. TCP and UDP specify the source and destination port numbers in their packet headers and that information, along with the source and destination IP addresses and the transport protocol (TCP or UDP), enables applications running on hosts on a TCP/IP network to communicate.

---

**Q2 a) What is sketching? Explain its role in Prototyping? (5m)**

- Sketching enables you to brainstorm, explore multiple ideas, define flows, and communicate with team members all while being quick and cheap.
- Prototyping enables you to get out of the building quicker, talk to users, validate assumptions, ensure expectations are meeting with stakeholders needs, while spending as little time as possible worrying about polish.



- There is a good chance that the first step we will take when working on our prototype will be to jot down some ideas or draw out some design ideas with pen and paper.
  - That is an important first step in exploring our idea and one we would like to extend beyond the strict definition to also include sketching in hardware and software.
  - What we mean by that is the process of exploring the problem space: iterating through different approaches and ideas to work out what works and what doesn't.
  - The focus isn't on the fidelity of the prototype but rather on the ease and speed with which we can try things out.
  - For the physical design, that could mean digging out childhood LEGO collection to prototype the mix of cogs and three-dimensional forms, or maybe attacking some foam core or cardboard with a craft knife.
  - Benefits of Sketching and Prototyping are:
    - Valid assumption
    - Discover problems early
    - Brainstorm Ideas
    - Design more Iterations
    - Conduct early usability testing
    - Cheaper and Faster to implement
    - Shorten the feedback loop
    - Help communication between team
- 

**Q2 b) "Open source has a competitive advantage". Discuss. (5)**

- Open source allows computer source code to be shared and modified free of charge by other users or organizations under a licensing agreement.
- In other words, it is a piece of software which can evolve and be iterated upon by other developers anywhere in the world.
- In case of open source, the software is evolved many times and during this evolution, it takes many twists and turns and becomes entirely different than its original version.

- For example: Android
  - Using open source is often a no-risk way of getting software that has been tested, improved and debugged by many eyes.
  - Secondly, using open source aggressively gives your product the chance to gain mindshare.
  - Companies are increasingly adopting open source software as a way to gain competitive advantage rather than just a cost-savings measure.
  - Open source software has been used for basic infrastructure, but organizations are now using it to build better software to support core business activities, such as data management and integration, application development etc.
  - Open source becomes better and stronger the more we apply it, contribute to it, and invest in its communities.
  - So we must apply our knowledge and efforts to ensure that open source can continue to prosper and serve as a source of competitive advantage for businesses and developers well into the future.
- 

**Q2 c) How can one tap into the community for promoting IoT devices? Explain. (5)**

- Local decision-making is sometimes considered a puzzle by research ecologists, resource managers, and policy researchers. The eternal hope is to find that model or concept that provides the “right” information to support local environmental decisions. Researchers have hypothesized that ecosystem services could be that concept.
- The Mid-Continent Ecology Lab is currently investigating that hypothesis through a systematic investigation of local values of ecosystem services, as well as the decision making process. We argue that “local” decisions are multi-dimensional and composed of overlapping and sometimes oppositional processes.

- Governmental bodies, non-profit organizations, and citizens all have expertise and knowledge to contribute to the process. This particular presentation will focus on one of these dimensions namely the citizen mobilization around the revitalization effort.
  - We present preliminary findings from research on the diverse efforts that citizens have undertaken in response to the revitalization efforts in the St. Louis River Corridor and the ecosystem services values that the underlie their activities. The contents of this abstract neither constitute nor necessarily reflect US PEA policy.
- 

**Q2 d) With the help of an example explain the process of Scaling up the electronics. (5)**

- Scaling up electronics is the process where a circuit is built from initial testing through prototype to a finished PCB.
- The starting point for prototyping is usually a Breadboard. This lets you push-fit components and wires to make up circuits without requiring any soldering and therefore makes experimentation easy.
- When we are happy with how things are wired up its easy to solder the components, which may be sufficient to make the circuit more permanent and prevent wires from getting away.

**For example:** Let us consider an evolution of the part of Bubblino circuit, from initial testing through prototype, to finished PCB.

- The first step in creating a circuit is generally to build it up on a breadboard, this way we can easily reconfigure things as we decide exactly how it should be laid out.
- When we are happy with how the circuit works, soldering it onto a stripboard will make the layout permanent. This means you can stop worrying about any wire getting loose, and if we are going to make only one copy of the circuit, then it might be as far as we need to take things.
- If we need to make copies of the circuit, or if we want a professional finish, we can turn our circuit into a PCB, this

makes it easier to build up the circuit because the position of each component will be labelled, there will be holes only where the components go, and there will be less chance of short circuits because the track between components will be protected by the solder resist.

---

**Q2 e) Explain the following IoT devices built with Arduino:**

**i) The Good Night lamp. ii) Botanicals iii) Baker treat (5)**

**i) The Good Night Lamp**

- The idea of Good Night Lamp was initially brought up by Alexandro Deschamps Soncino. He came up with an idea of internet connected table or bedside lamp.
- A simple consumer device, this lamp would be paired with another lamp anywhere in the world, allowing it to switch the other lamp on or off, and vice versa.
- Because light is integrated into our daily routine, seeing when our loved ones turn, for example, their bedside lamp on or off gives us a calm and ambient view onto their lives.
- The product consists of a “big lamp” which is paired with one or more “little lamps”. The big lamp has its own switch and is designed to be used like a normal lamp. The little lamps, however, don’t have switched but instead reflect the state of the big lamp.
- Adrian and the rest of the team’s familiarity with Arduino led to it being an obvious choice as the prototyping platform. In addition, as the lamps are designed to be a consumer product rather than a technical product, and are targeted at a mass market, design, cost, and ease of use are also important.
- The Arduino platform is simple enough that it is possible to reduce costs and size substantially by choosing which components you need in the production version.
- A key challenge in creating a mass-market connected device is finding a convenient way for consumers, some of

whom are non-technical, to connect the device to the Internet. Even if the user has Wi-Fi installed, entering authentication details for your home network on a device that has no keyboard or screen presents challenges.

- As well as looking into options for the best solution for this issue, the Good Night Lamp teams are also building a version which connects over the mobile phone networks via GSM or 3G. This option fits in with the team's vision of connecting people via a "physical social network", even if they are not otherwise connected to the Internet.

## **ii) Botanicals**

- Botanicals are collaboration between technologists and designers that consists of monitoring kits to place in plant pots.
- The Botanicals kits then contact the owner if the plant's soil gets too dry. The project write-up humorously refers to this as "an effort to promote successful inter-species understanding" and as a way of translating between a plant's communication protocols to human protocols, such as telephone, email, or twitter.
- The original project used stack Arduino controllers, although the kits available for sale today use the AT mega 168 microcontroller with a custom board, which remains Arduino-compatible, and the programming is all done using the Arduino IDE. To match the form factor of the leaf-shaped printed circuit board, the device uses a Wiz Net Ethernet chip instead of the larger Arduino Ethernet Shield.

## **iii) Baker treat**

- The Baker Tweet is effectively a physical client for Twitter designed for use in bakery.
- A baker may want to let customers know that a certain product has just come out of the ovens-fresh bread, hot muffins, cupcakes laden with icing-yet the environment he would want to tweet from contains hot ovens, flour dust, and sticky dough and batter, all of which would pay havoc with the electronics, keyboards, and screen of

- a computer, tablet, or phone.
- Staff of design agency Poke in London wanted to know when their local bakery had just produced a fresh batch of their favorite bread and cake, so they designed a proof of concept to make it possible.
  - Because Baker Tweet communicates using Wi-Fi, bakeries, typically not built to accommodate Ethernet cables, can install it. Baker Tweet exposes the functionality of Twitter in a “bakery-proof” box with more robust electronics than a general-purpose computer, and a simplified interface that can be used by fingers covered in flour and dough.
  - It was designed with an Arduino, an Ethernet Shield, and a Wi-Fi adapter. As well as the Arduino simply controlling a third-party service, it is also hooked up to a custom service which allows the baker to configure the messages to be sent.
- 

**Q3 a) Explain the sketch iterate and explore process in prototyping. (5)**

- Sketching enables you to brainstorm, explore multiple ideas, define flows, and communicate with team members all why being quick and cheap. Iterative design is a design methodology based on a cyclic process of prototyping, testing, analyzing, and refining a product or process.
- A prototype is a draft version of a product that allows you to explore your ideas and show the intention behind a feature or the overall design concept to users before investing time and money into development.
- Doing a large amount of introductory jobs doesn't mean that you won't do extra research and gathering of possible beginning points when you do eventually sit down with a particular estimate in mind.
- To start with, you're looking for an extensive search across the problem area. The motive is to get to clutch with as many aspects of the design as possible, rather than drilling down into

one specific possible result. Pushing beyond the obvious solutions forces you to look at things differently and increases the likelihood that you'll have a good design among your options.

- Use whatever tools make most sense to help with the idea triggering and investigation. You might use a mood board—a whiteboard where you jot down thoughts and sketches over a few days—or a notebook that you doodle sketches.
- They only need to capture and convey your ideas. The more sketching you do, the better they will get.
- Then give it or show it to some of the people who might use the finished item to find out how they interact with it.
- Arguably, in the early stages of a design, you can never do too much iterating through ideas and trying out different approaches to solving the problem.
- If the idea warrants it (and maybe even if it doesn't), don't be afraid to take your sketching into three dimensions. Mock up different designs with modelling clay or LEGO or some of the other methods we cover in this chapter. Try out different sizes and see how the changes in dimensions affect the feel or the look of the design.
- Maybe even combine the approaches here with the things you learnt in for prototyping the software and electronics and lash up a rough-and-ready prototype that you can try out properly. Then give it or show it to some of the people who might use the finished item to find out how they interact with it.

For example, the evolution of the design for the Good Night Lamp (which we featured in a case study in the last chapter). The original design was a more traditional lamp shape, but in a design workshop, the team batted around a range of ideas with the help of a purely functional prototype. They realized that a design echoing the shape of a house better conveyed the core concept of connecting loved ones in their homes.

- The key lesson is to use these techniques to experiment with different possibilities and learn which features of which designs are best. This approach allows you to synthesize the results into a coherent final design.

---

**Q3 b) Write a short note on laser cutters. (5)**

- Laser cutting is a technology that uses a laser to cut materials, such as industrial manufacturing, glass jars, metallic things etc.
- But it is also starting to be used by schools, small businesses, and hobbyists.
- Laser cutting works by directing the output of a high-power laser most commonly through optics.
- A commercial laser for cutting materials involved a motion control system to follow a CNC or G-code of the pattern to be cut onto the material.
- The focused laser beam is directed at the material, which then either melts, burns, vaporizes away, or is blown away by a jet of gas, leaving an edge with a high- quality surface finish.
- Industrial laser cutters are used to cut flat-sheet material as well as structural and piping materials.
- The laser cutting machine was first used on 1965, to do holes in diamond dies.
- The first machine was made by Western Electric Engineering Research Center.
- Laser printer is very useful to have in your workshop.
- 3D printer can produce more complicated parts, but the simpler design process, greater range of materials which can be cut at faster speed makes the laser cutter a useful product.
- Choosing a laser cutter:
  - **The size of Bed:** This is the surface where the sheet of material is kept at the time of cutting. Bigger bed can be more cost effective.
  - **Power of Laser:** More power of the cutter can cut thicker sheets. E.g. 40W laser can cut 10mm thick acrylic and 60W laser can cut 25mm thick acrylic.

---

**Q3c) What is milling? Explain. (5)**



- Milling is the process of machining using rotary cutters to remove material by advancing a cutter into a work piece.
- This may be done varying direction on one or several axes, cutter head speed, and pressure.
- Milling covers a wide variety of different operations and machines, on scales from small individual parts to large, heavy-duty gang milling operations.
- It is one of the most commonly used processes for machining custom parts to precise tolerances.
- Milling can be done with a wide range of machine tools.
- The original class of machine tools for milling was the milling machine (often called a mill).
- After the advent of computer numerical control (CNC) in the 1960s, milling machines evolved into machining centers: milling machines augmented by automatic tool changers, tool magazines or carousels, CNC capability, coolant systems, and enclosures.
- Milling centers are generally classified as vertical machining centers (VMCs) or horizontal machining centers (HMCs).
- The integration of milling into turning environments, and vice versa, began with live tooling for lathes and the occasional use of mills for turning operations.
- This led to a new class of machine tools, multitasking machines (MTMs), which are purpose built to facilitate milling and turning within the same work envelope.

### **Milling Process:**

- Milling is a cutting process that uses a milling cutter to remove material from the surface of a work piece.
- The milling cutter is a rotary cutting tool often with multiple cutting points.
- As opposed to drilling, where the tool is advanced along its rotation axis, the cutter in milling is usually moved perpendicular to its axis so that cutting occurs on the circumference of the cutter.
- As the milling cutter enters the work piece, the cutting edges

(flutes or teeth) of the tool repeatedly cut into and exit from the material, shaving off from the work piece with each pass. The cutting action is shear deformation; material is pushed off the work piece in tiny clumps that hang together to a greater or lesser extent (depending on the material) to form chips.

- This makes metal cutting somewhat different from slicing softer materials with a blade.

---

**Q3 d) What are the legalities associated with scrapping? (5)**

- Most of the companies have sample of relevant data but they don't have knowledge to make it available as API's.
- From Facebook example you understand that the computer is only supposed to run the browser and navigate it through UI elements which can be delicate but making computer do more work is not impossible.
- This is known as "screen - scrapping"
- Screen-scrapping is not useful for website as it may break the terms and conditions that a website can follow.
- For example the Google won't allow to screen scrap its search bar but will provide a API for the same. Even if you don't follow the legal rules the basic terms and conditions that are put by the company like Google should be followed else the company will exclude you from all other services that is provides that me lead to in convenience.
- Rest of the data is protected with various security techniques like copyright, authenticating users to database rights and many more.
- There are always alternatives for all the information that is available. For example you can use Microsoft Outlook rather than using Gmail or you can use OpenStreetMap rather than using Google Maps.

---

**Q3e) What is comet? Explain (5)**

- Comet is an umbrella name for a set of technologies developed to get around the inefficiencies of polling. As with many technologies, many of them were developed before the “brand” of Comet was invented; however, having a name to express the ideas is useful to help discuss and exchange ideas and push the technology forward.

1. Long Polling (Unidirectional):

- The first important development was “long polling”, which starts off with the client making a polling request as usual. However, unlike a normal poll request, in which the server immediately responds with an answer, even if that answer is “nothing to report”, the long poll waits until there is something to say.
- This means that the server must regularly send a keep-alive to the client to prevent the Internet of Things device or web page from concluding that the server has simply timed out
- Long polling would be ideal for the case of Where Dial: the dial requests to know when the next change of a user’s location will be. As soon as Where Dial receives the request, it moves the dial and issues a new long poll request.

2. Multipart XMLHttpRequest (MXHR)(Unidirectional):

- When building web applications, it is common to use a JavaScript API called XMLHttpRequest to communicate with the web server without requiring a full new page load.
- From the web server’s point of view, these requests are no different from any other HTTP request, but because the intended recipient is some client-side code, conventions and support libraries (both client- and server-side) have developed to address this method of interaction specifically.
- Many browsers support a multipart / x-mixed-replace content type, which allows the server to send subsequent versions of a document via XHR.

### 3. HTML5 WebSockets (Bidirectional):

- Traditionally, the API used to talk directly to the TCP layer is known as the sockets *API*. When the web community was looking to provide similar capabilities at the HTTP layer, they called the solution WebSockets.
- Although WebSockets are currently a working draft in the HTML5 spec, they seem to have traction in modern browsers, servers, and other clients
- For example, there is a (partial) implementation for the Arduino platform. WebSockets have the benefit of being bidirectional. You can consider them like a full UNIX socket handle that the client can write requests to and read responses from.
- This might well be the ideal technology for the task timer. After a socket is established, the timer can simply send information down it about tasks being started, modified, or cancelled, and can read information about changes made in software, too.

### 4. Implementations:

- The options described in the preceding section seemed to us to have most traction currently; however, as a fast-changing area with no absolute consensus as yet, the actual details of transports and limitations are bound to change. It is worth paying attention to these transports as they develop. The Wikipedia page on Comet is a useful starting point for tracking the current state of play.
- Let's look at support for these techniques on the three main platforms that you may need to consider for an Internet of Things application: the browser web app (if applicable), the microcontroller itself, and the server application.
- On the browser side, it is often possible to abstract the actual transport using a library which chooses which method to connect to the server. For example, it might use WebSockets if available;

otherwise, it will fall back to XMLHttpRequest or long polling.

- This capability is useful because each web browser currently has varying levels of support for the different techniques. There are well-known Comet libraries for jQuery and for Dojo.
- There are also libraries for the microcontroller; however, they tend to support only one scheme. For example, several dedicated WebSockets libraries are available for Arduino. In fact, the fallbacks to different methods of interchanging data aren't really needed on the Arduino.
- Unlike the case of a desktop web app, with Arduino you don't have to worry about the users having different browsers because you'll be providing the firmware for the device.

#### 5. Scaling:

- An important consideration is that all these Comet techniques require the client to have a long-term connection with the server. For a single client, this is trivial. But if there are many clients, the server has to maintain a connection with each of them.
- If you run a server with multiple threads or processes, you effectively have an instance of the server for each client. As each thread or process will consume system resources, such as memory, this doesn't scale too many clients.
- Instead, you might want to use an asynchronous web server, which looks at each client connection in turn and services it when there is new input or output.
- If the server can service each client quickly, this approach can scale up to tens of thousands of clients easily. There is a problem that each process on a typical UNIX server has a maximum number of sockets, so you are restricted to that number of simultaneous clients.

---

**Q3 f) Explain HTML5 web socket**

**(5)**

- Web Socket is a web technology that provides full-duplex communication channels over a single TCP connection. A full-duplex communication is the communication system that allows simultaneous bidirectional communication. A telephone conversation is a good example of full-duplex communication where in both parties can speak and hear at the same time.
- The API used to talk directly to the TCP layer is known as the sockets API. When the web community was looking to provide similar capabilities at the HTTP layer, they called the solution Web Sockets.
- Web Sockets are currently a working draft in the HTML5 specification, they are embedded in modern browsers, servers, and other clients.
- Web Sockets have the benefit of being bidirectional. You can consider them like a full UNIX socket handle that the client can write requests to and read responses from.
- This might well be the ideal technology for the task timer. After a socket established, the timer can simply send information down it about tasks being started, modified, or cancelled, and can read information about changes made in software, too.
- Because Web Sockets are new and push the HTTP protocol in a slightly unorthodox direction, they are known to have some issues with proxy servers. This situation should change as the proxies currently broken in this respect are fixed to be aware of Web Sockets.
- One of the more unique features WebSockets provide is its ability to traverse firewalls and proxies, a problem area for many applications. Comet-style applications typically employ long- polling as a rudimentary line of defense against firewalls and proxies.

- The technique is effective, but is not well suited for applications that have sub 500 millisecond latency or high throughput requirements.
  - A WebSockets detects the presence of a proxy server and automatically sets up a tunnel to pass through the proxy.
  - WebSockets—like other pieces of the HTML5 effort such as Local Storage and Geolocation—was originally part of the HTML5 specifications.
- 

**Q4 a) Explain different types of memory. (5)**

Types of memory

**ROM (Read-only Memory):**

- ROM is non-volatile memory so always retains its data.
- Read-only memory is used to store the hard-coded information of chips and can only be read afterwards.
- It is used to store only the executable program code and any data which is fixed and never changes.
- ROM's are the least flexible memory.
- Creating memory with ROM is the cheapest way.
- It is less flexible from flash memory.
- ROM needs power to store information and no data lose happen whenever the power failure occurs.

**Flash:**

- Flash is a semi-permanent type of memory which provides all the advantages of ROM.
- Flash doesn't need power to store information, so its contents can survive after the circuit being unplugged.
- The content of flash can be rewritten a maximum number of times.
- The reading speed of information is not much different as of speed of RAM or ROM.
- In flash writing process takes few processor cycles, which means it's best to store the program executable data or important data that has been gathered.

### **RAM (Random-access Memory):**

- RAM is also called as read write memory or the main memory or the primary memory.
  - RAM is volatile memory as the data loses when power is turned off.
  - Random-access memory mostly used for its speed to access the data.
  - RAM requires power to retain its contents.
  - RAM is used as working memory for the systems, the place where the things are stored while being processed.
- 

**Q4b) With the help of examples, compare stack and heap. (5)**

#### **1. Stack:**

- New items which are added to the stack go on the top, and items can be removed only in strict reverse order, so the first thing to be removed is the last item that was placed onto the stack.
- This arrangement makes it easy for the processor to keep track of where things are and how much space is being used because it has to track only the top of the stack.
- Consequently, the stack is really only useful for
  - Items that aren't going to survive for long periods of time.
  - Items that remain in constant use, from the beginning to the end of the program
- The downside to this approach is that if you're finished with a particular variable, you can release the memory used for it only when you can remove it from the stack, and you can do that only when everything added since it was allocated is removed from the stack, too.
- Global variables, which are always available, are allocated first on the stack. After that, whenever the path of execution enters a function, variables declared within it are added. The parameters to the function get pushed onto the stack immediately, while the other variables are pushed as they are encountered. Because all the variables



within a function are available only to code inside it, when you reach the end of that function, all those parameters and variables are ready to be discarded. So, the stack gets unwound back to the same size it was just before control passed to the function.

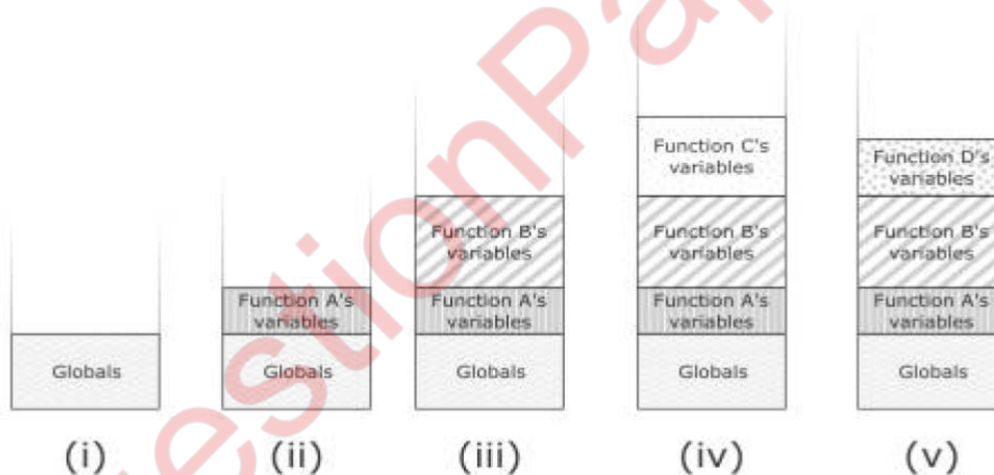
- Example:

```
//Global Variables  
function A  
{  
variable A1;  
variable A2;  
call B();  
}  
function B  
{  
variable B1;  
variable B2;  
variable B3;  
call C();  
call D();  
}  
function C  
{  
variable C1;  
variable C2;  
}  
function D  
{  
variable D1;  
}  
call A();
```

Before function A is called, the stack looks like state (i).

- As execution moves into function A, its variables are added to the stack (ii).
- Function A then calls function B, resulting in its variables being added to the stack (iii).

- Inside function B, first function C is called, resulting in its variables being added to the stack (iv).
- When execution returns from function C, its variables are removed from the stack, taking you back to stack (iii).
- Then function D is called, so its variables are pushed onto the stack instead (v).
- Then execution returns to function B, with D's variables removed (iii).
- And back to A, removing B's variables (ii).
- And, finally, you leave function A, dropping back to just the global variables being defined (i).



## 2. Heap:

- Heap allows the allocation of chunks of memory at any time.
- The heap is a bit like the seating area of a train where you fill up the seats strictly from the front and have to keep everyone who is travelling as a group in consecutive seats.
- To begin, all the seats are empty. As groups of people arrive, you direct them to the next available block of seats.
- But two possible problems exist,
  - First, you might simply have more people to fit on the train than there are seats (this is the same

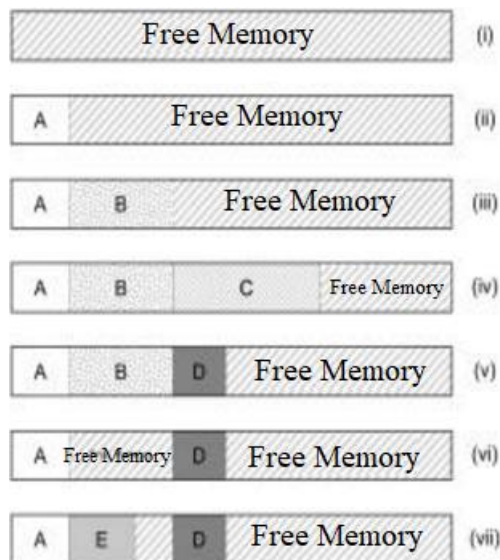
problem as running out of memory).

- The second problem is more subtle: though you theoretically have enough free seats for the next group of passengers, those free seats are spread across the train and aren't available in a continuous block of free seats. This last situation is known as memory fragmentation.

- Example:

```
create object A (size 20 bytes)  
create object B (size 35 bytes)  
create object C (size 50 bytes)  
// do some work that needs object C  
delete object C  
create object D (size 18 bytes)  
// do more work with objects B and D  
delete object B  
create object E (size 22 bytes)
```

- At the start of execution, the heap will be empty (i).
- Object A is added to the heap (ii), taking up 20 bytes of space.
- Object B is added to the heap (iii), consuming a further 35 bytes straight after the space for object A.
- Object C is added to the heap (iv), adding 50 bytes to the heap right after object B.
- Object C is no longer needed and is deleted; releasing the space it consumed on the heap and taking us back to heap (iii).
- Object D is created and takes up 18 bytes of the space just vacated by object C (v).
- Now object B is finished with and deleted. As other code might be relying on the position of object D, we can't move it, so there's now a free space between objects A and D (vi).
- Object E is created. It requires 22 bytes of space, which means it will fit in the hole left by object B (vii).



**Q4 c) What are libraries? Explain with examples (5)**

- When you are developing software for server or desktop machines you need to also develop array of libraries and framework to make use of software easy.
- In embedded system it becomes bit tricky and becomes better with time as it provides system on chip functionality.
- The other component is micro controller which are resource constrained which pulls mainstream operating system libraries and its code.
- You can use existing code to develop your version but if unnecessary memory allocations are been done by the code it would lead to memory wastage so it's better to start writing your version from scratch keeping the limitation of micro controller in mind instead of simply wasting the memory.
- Following are few commonly used libraries:
  - **IwIP**
    - Also known as LightWeight IP. It is a stack of TCP/IP which runs in condition where there are fewer resources.
    - Requires less memory of about 10KB of RAM and 40KB of ROM/Flash. Arduino Wi-Fi uses such type of library.
  - **uIP**
    - uIP, or micro IP is TCP/IP stack for smallest possible

system. It can even run on system with only a couple of kilobytes of RAM.

- It does not have any buffer which stores incoming and outgoing packets which are not acknowledged.

➤ **uClibc**

- uClibc is a version of standard GNU C library for embedded Linux system.
- The source code of the library can be changed and to implement the change you only need to recompile the code.

➤ **Atomthreads**

- Atomthreads is lightweight real-time scheduler for embedded systems.
- It can be used when the code gets complicated and you want to make more than one task to execute together, scheduler plays an important role here as it switches between the tasks quickly which looks pretty simple as that of multitasking computer.

---

**Q4 d) Define business model .Explain different factors in the definition. (5)**

- Business Model describes the rationale of how organization creates, delivers, capture value in economic, social, cultural or other contexts And the process of business model construct and modify is also called business model innovation.
- The main motive of developing a model for your business is to get a proper idea of what you are trying to do and whether your business is delivering the same what you expected out of it.
- Even if you don't follow the formal way of developing the model, the one who thinks about establishing a business as thought in details about things like whether it would be affordable, what business would actually be about and can profit be achieved out of it.
- The business model definition brings together a number of factors:
  - A group of people.








- People who are going to use the product.
- The need of those customers.
  - What the customer or the people who are going to be potential user of the product expect out of the product.
- A thing that your business can do to meet those needs.
  - Strategies and tactics that can be developed to meet these needs.
- A success criterion such as making a profit.
  - Most of the ideas that looked great and theoretically possible on paper turned out to be practically impossible or high budget. Making all possible combination and trying all possible ideas will help your product to grow and become strong or else it will let you know where your product stands. Business models can also be helpful to get more people involved in this process, this people can be your stakeholders which includes employees, customers, etcetera.
- Organizational practices that help to achieve this goal.
  - Different activities should be practiced so that everyone involved in the project should believe that the product that is been developed is capable enough to survive in the market and can also gain profit from the market.
- And to be able to carry on doing so, sustainably.

---

**Q4 e) With the help of a diagram, explain business model canvas.**

**(5)**

- One of the most popular templates for working on a business model is the Business Model Canvas by Alexander Osterwalder and his startup, the Business Model Foundry. The canvas is a Creative Commons–licensed single-page planner.
-

<p><b>Key Partners</b></p> <p>Identify your company's key partners. This can consist of important suppliers in your supply chain. What <b>key resources</b> does the company receive from these partners? What <b>key activities</b> are performed by these partners? Think about why your company works with these key partners and the motivations behind them.</p> 	<p><b>Key Activities</b></p> <p>What specific key activities are necessary to deliver your <b>value proposition</b>? What activities set your company apart from others? Consider how your company's unique differences in its <b>revenue streams</b>, distribution <b>channels</b>, or <b>customer relationships</b>. Do you need to procure specific niche resources? Do you need to streamline to keep costs and prices low?</p> 	<p><b>Value Propositions</b></p> <p>Identify the core value the company provides to customers. What exactly is the company trying to give to customers? What problem is your company trying to solve and what needs are your company satisfying? How do you offer something different that satisfies the demands of your <b>customer segments</b> (e.g. price, quality, design, status)?</p> 	<p><b>Customer Relationships</b></p> <p>What type of relationship do you have with your customers? How do you interact with customers and how does this differ amongst <b>customer segments</b>? Do you communicate frequently with your customers? How much support does your company provide?</p> 	<p><b>Customer Segments</b></p> <p>Identify who is your <b>value proposition</b> targets. Who are you creating value for? Who are your most important customers? What are they like? What do they need? What do they enjoy? What is the customer market like? Are you targeting a small niche community or a mass market?</p> 
<p><b>Cost Structure</b></p> <p>Identify the key costs in your company's business model. What are the major drivers of costs? How do your <b>key activities</b> and <b>key resources</b> contribute to the cost structure? How do your costs relate to your <b>revenue streams</b>? Are you properly utilizing economies of scale? What proportion of costs are fixed and variable? Is your company focused on cost optimization or value?</p> 		<p><b>Revenue Streams</b></p> <p>Identify the ways your <b>value proposition</b> generates money for your business. Does your company have multiple methods of generating revenue? What is the pricing strategy for the products offered by your company? Through what channels do your customers pay? Does your company offer multiple forms of payment?</p> 		

- At first sight, it looks as though each box is simply an element in a form and the whole thing could be replaced by a nine-point checklist. However, the boxes are designed to be a good size for sticky notes, emphasizing that you can play with the ideas you have and move them around. Also the layout gives a meaning and context to each item.
- Let's look at the model, starting with the most obvious elements and then drilling down into the grittier details that we might neglect without this kind of template.
- At the bottom right, we have Revenue Streams, which is more or less the question of "how are you going to make money?" we used to start this chapter. Although its position suggests that it is indeed one of the important desired outputs of the business, it is by no means the only consideration. The central box, Value Propositions, is, in plainer terms, what you will be Producing that is, your Internet of Things product, service, or platform. The Customer Segments are the people you plan to deliver the product to. That might be other makers and geeks.
- The Customer Relationships might involve a lasting communication between the company and its most passionate customers via social media. This position could convey an

advantage but may be costly to maintain. Maintaining a “community” of your customers may be beneficial, but which relationships will you prioritize to keep communicating with your most valuable customer segments?

- Channels are ways of reaching the customer segments. From advertising and distributing your product, to delivery and after-sales, the channels you choose have to be relevant to your customers.
- On the left side, we have the things without which we have no product to sell. The Key Activities are the things that need to be done. The Thing needs to be manufactured; the code needs to be written. Perhaps you need a platform for it to run on and a design for the website and the physical product.
- Key Resources include the raw materials that you need to create the product but also the people who will help build it. The intellectual resources you have are also valuable, as are the finances required to pay for all this.
- Of course, few companies can afford the investment in time and money to do all the Key Activities themselves or even marshal all the Key Resources. We will need Key Partners, businesses that are better placed to supply specific skills or resources, because that is their business model, and they are geared up to do it more cheaply or better than you could do yourself. Perhaps you will get an agency to do your web design and use a global logistics firm to do your deliveries. Will you manufacture everything yourself or get a supplier to create components or even assemble the whole product?
- The Cost Structure requires you to put a price on the resources and activities you just defined. Which of them are most expensive? Given the costs you will have, this analysis also helps you determine whether you will be more cost driven (sell cheaply, and in great volume via automation and efficiency) or more value driven (sell a premium product at higher margins, but in smaller quantities).

---

**Q4 f) What is venture capital? How can one exit?**

**(5)**



- Venture capital is finance that investors provide to startup companies and small businesses that are believed to have long-term growth potential. However, it does not always take just a monetary form, it can be provided in the form of technical or managerial expertise.
  - Venture capital means risk capital. The risk visualized may be very high, may be so high as to result in loss or very loss so as to result in high gains.
  - The main objective of venture capital is not to gain interest but capital gain.
  - The venture capital investment is made when a venture capitalist buys shares of such a company and becomes a financial partner in the business.
  - The Venture capital typically comes from institutional investors and high net worth individuals and is pooled together by dedicated investment firms.
  - It is the money provided by an external investor to finance a new, growing, or troubled business.
  - **Advantages of Venture capital investing:**
    - It provides large sum of equity finance
    - Economic growth
    - It is able to bring wealth and expertise to your company
    - It also has wide network of contacts
    - It provides additional funds
  - **Disadvantages of venture capital investing:**
    - You must generate the cash needed to make the agreed payments of capital, interest and dividends. This can create great financial pressure.
    - You will have to agree to certain restrictions as part of the deal, such as the amount you are paid and your involvement with other businesses and you will usually need your investors consent to major decisions.
- How can one exit?**
- An exit does not mean that you have to leave the company. In fact, in many cases, the exit terms may require that you stay for a maximum of one to three years after acquisition in order

to ensure continuity and value for the buyer. An exit strategy is a “method by which a venture capitalist or business owner intends to get out of an investment that he or she made”. Because your investors will want a return, your long-term goal can’t just be to make your company successful but to do it in such a way as to pay back the investment.

Typically, there are only two exits:

- **You get bought by a bigger a company:**
  - In this case, the buyer buys out the investors that is, the buyer pays the investors value of their percentage equity of their perceived valuation of the worth of the company.
  - Founding members of the company often transfer to the purchasing company, as they constitute one of the company’s principal resources.
  
- **You do an IPO(Initial Public Offering)-that is, float on the stock market:**
  - This involves new shares being issued and sold to the stock market. Although this option “dilutes” the value of the shares already issued, the existing holders are able to the sell their shares on the market too, to get back their investment, or to retain the shares if they believe that the shares will grow in value.

---

**Q5 a) Explain in details the process of designing the kits. (5)**

- The first step towards selling your idea as a product is to provide it as a kit. Many a time we underestimate the work and start buying it but we will not be able to select the right component and putting together with a step-by-step guide.
- Most kits tend to provide only the electronics and software for a particular application rather than any physical housing. The reason partly comes down to the difficulty, for a cottage kit-building industry, of sourcing custom plastic components.
- The main market for such kits is others in the maker community; these makers will perhaps prefer to combine the kit into a project of their own. However, with the growing

accessibility of 3D printers and laser-cutters, it is becoming vastly more feasible to provide housing and other physical components even for kits.

- Kits tend to piggyback on existing microcontrollers and often take the form of a standard format plug-on board for example shield in the Arduino ecosystem or a cape on the Beagle Bone.
- It reduces the support overhead for the kit provider; either user will already be familiar with the platform or if not plenty of assistance will be available elsewhere to cover the basics of getting up and running. The kits documentation can then focus on just what is specific in building the project.
- If you haven't been building electronics for long, designing a PCB is likely to involve a number of new terms and concepts. Similarly, you can make the PCBs a few different ways after they are designed.
- For example, we like the tutorials that Adafruit put together for each of its products; the Adafruit Motor Shield for Arduino is typical.
- One should create a spreadsheet (or at least a list) of all the items that make up the product, along with their cost to buy. One should list every single electronic component, connector, cable, PCB, case, and so on, in addition to the packing box you'll ship it in and the time taken to put things together. This list is called the bill of materials (BOM), and it forms the starting point for all your costings and prices.
- The BOM gives you the marginal cost for your product—that is, how much extra it would cost to make an additional item when you're up and running with production.
- A good starting point for working out your price (how much you will charge consumers) is to take the total cost of the BOM and multiply it by 4 or 5. That calculation gives you a margin to cover that item's portion of the fixed costs and also some profit. It also provides enough margins for resellers to cover their fixed costs and make some profit.
- However, it can also add to the support overhead, because you will need to deal with remotely debugging your customers' issues, which may be down to their poor soldering rather than

defects in your work. These problems can all be resolved by moving on a step towards a consumer product and selling fully

Assembled PCBs, populated with all the components.

---

**Q5 b) Write a short note on mass-producing the case and other fixtures. (5)**

- A good rule is to keep down cost production is to minimize amount of time a person has work on each item. Machines will be tend to be cheaper than people.
  - The most common method of mass production: injection molding of plastic. The process involves injecting molten plastic which will be converted into desired shape.
  - After the plastic has cooled sufficiently, the mold is separated out successfully and other parts of injection will collect in collection bin.
  - The whole cycle takes less time than 3-D printing, which means lot of parts can easily churned at lower cost per part.
  - Case study: BERG's little printer. The little printer, made by London design firm BERG. Initially it was called social letterbox. With the idea that it will be socially able to share printed thing among when family is not close to each other.
  - Then printer is connecting to computer, the printer has its own network connection and intelligence.
  - It is a two-stage process for part of the case. It is injection-molded without the holes, which gives the correct finish, and then drilled to provide the holes. This process is expensive but it results in the higher quality finish.
  - BERG cloud bridge, a box with both ZigBee wireless and Ethernet which you plug into an Ethernet port on your network to talk to Internet.
  - Printer allows communicating with BERG cloud server software, which provides for both the delivery of 'publication' to printer and the user-facing website.
-

**Q5 c) Discuss the issues in scaling up the software for large scale IOT devices. (5)**

The following are the issues in scaling up the software for large scale IOT devices:

Deployment:

- The process of moving software from machine to where it is to be executed is known as deployment of the software.
- In case of embedded system, the software will be burnt on the controller during manufacturing itself.
- The software for an online service will tend to run in a single location and is more vulnerable to malicious inputs that might cause it to stop working. You should also focus on the topic on how the software will be regularly updated.
- If you are using a hosted service such as Heroku, there will be simple, standard ways to do this. If you are running your own dedicated web server or perhaps a virtual machine such as an Amazon EC2 instance, there are many solutions, from shell scripts using SCP, rsync, or git to copy code, to deployment frameworks such as Capistrano.

Correctness and Maintainability:

- While dealing with an embedded environment, the software should do what it has meant to be.
- Example, it shouldn't happen the coffee machine makes cappuccinos when the customer asked for a latte.
- So the software should do its given task efficiently and safely. Testing your code before it is deployed is an important step in helping to avoid such a situation.
- The embedded code in the device, however, is particularly important to test. Also you should focus on updating the software efficiently.

Security: Following are some of the more important guidelines:

- Servers should be kept up-to-date with the latest security patches; user passwords should never be stored in plain text. If your database were ever compromised, an attacker could easily log in as any user. Use of Hashing is encouraged.

- Never simply trust user input, be alert with spoofing attack and attacks like SQL injection.
- Be aware of cross-site request forgery (CSRF) attacks from other malicious or compromised websites.

Performance:

- Whenever you think of scaling up software see to it that the system performance is not degraded.

User Community:

- Once the software is made available, you should be in state to provide a good customer service. While few large companies can boast Apple's famous commitment to "insanely great customer service", a small, focused startup can often match them for responsiveness and enthusiasm.
- If not then at least keep good online tutorials, forums, blogs and chat room that lets users support each other.

---

**Q5 d) Discuss the advantage and disadvantages of technology. (5)**

**Advantages of Technology:**

- Improves communication: Many businesses are using various business communication technologies to change the way their employees interact and communicate while at work.
- Encourages Innovation and Creativity: Workers can use different business technologies to create innovative business ideas which can be used in business growth and expansion.
- Improves on Human Resource Management: It improves on the process of screening, recruiting and hiring new employees. Many human resource managers are using internet to advertise job openings
- Saves Time: Technology can be used to automate various tasks at work, this automation will guarantee efficiency and will also increase on production at work.

**Disadvantages of Technology:**

- Causes Distraction at Work: Their so many ways technology can distract employees at work. The use of social networks at work can cause so much distraction and it affects the productivity of employees.

- High Maintenance costs: It is expensive to buy technology, but it is also costly to maintain it.
- Makes employees lazy: Since most tasks are automated by technology, many employees become lazy at work, technology kills their creativity and skills. Affects Workplace Relationships: Employees communicate via cell phones, text messages, email or virtual video conferencing tools. This type of communication technology eliminates face-to-face communication.

---

**Q5 e) “The internet destroys the state”. Discuss (5)**

- The Internet is a massive open publishing platform, has been a disruptive force in regards to privacy.
- The privacy disrupts because everything we write might be visible to anyone online. From small details like what we ate in breakfast to the blog post about our work, everything is visible.
- Do we always want people to be able to see our data? .we might not but big companies and government might see our activities without our consent.
- The Canadian open government activist David Eaves has discussed that it is not only Iran or china which are intent on controlling the internet but also other countries which are democratic.
- The US, UK, Canada, France, and others have already enacted various laws to give the state and its favored corporation’s greater control over its citizens’ use of the Internet.
- But Eaves also suggested the other possibility is that “The Internet Destroys the State”. This might be hard to imagine but when we understand or tone down this idea a little bit, “the Internet” fighting back against an attempt by the state or corporations to co-opt it.
- Maybe this isn’t a world with no state – but lots of little states or maybe some new organizing mechanism, one which leverages local community identities, but can co-exist with a network of diffused but important transnational identities. Or maybe the organizing unit gets bigger so that greater resources can be called upon to manage new network-based threats.

- People still find this world harder to imagine. This is because so many of our assumptions suddenly disappear. If not the state, then what? Who or what protects and manages the internet infrastructure? What about other types of threats – corporate interests, organized and cyber-crime, etc.?
  - We can see real steps at disrupting control, however, in a slightly more prosaic but equally fascinating area.
- 

**Q5 f) What is environmental cost of Internet service for IoT device? What is the solution? (5)**

- Creating the object has a carbon cost, which may come from the raw materials used, the processes used to shape them into the shell, the packing materials, and the energy required to ship them from the manufacturing plant to the customer.
- It's easier than ever to add up the cost of these emissions: for example, using the ameeConnect API we can find emissions data and carbon costs for the life-cycle, use of different plastics, we might use for 3D printing or injection molding. Calculating the energy costs for manufacture is harder.
- The electronics contained in a thing have their own environmental cost. Buying PCBs locally or from a foreign manufacturer affects the carbon cost of shipping the completed units.
- Considering the potential cost savings, even a responsible manufacturer may find it reasonable to offset the extra carbon emissions. If our product needs to conform to RoHS legislation, then every single component that could be extracted from it must be RoHS compliant.
- In the digital world, moving data rather than physical objects is faster, is safer, and has a lower environmental cost. Of course, "data" doesn't exist in the abstract. The stone tablets, parchment scrolls, and libraries of paper books or microfiche that have historically been used to store analogue data always had their own environmental cost.
- Now, running the Internet has a cost: the electricity to run the routers and the DNS lookups, plus establishing the



infrastructure—laying cabling across the sea, setting up microwave or satellite links, and so on.

- As well as the cost of transferring the data across the Internet, running our own web server uses power. Many server hosting specialists now offer carbon-neutral hosting
  - Solutions like Instrumenting production lines, home energy usage, transport costs, building energy efficiency, and all other sources of efficiency might seem extreme, but it may be a vital, imperative task. Projects such as a carbon score for every company in the UK will help change attitudes, perhaps simply by gamifying the process of improving one's emissions, but also by having an objective measure that could, in future, be as important to a company's success as its credit score.
-