CMR
INSTITUTE OF
TECHNOLOGY

Internal Assesment Test - I

Sub:	SOFTWARE ENGINEERING							Code:	15CS42
Date:	27 / 03 / 2017	Duration:	90 mins	Max Marks:	50	Sem:	4(A,B,C)	Branch:	CSE

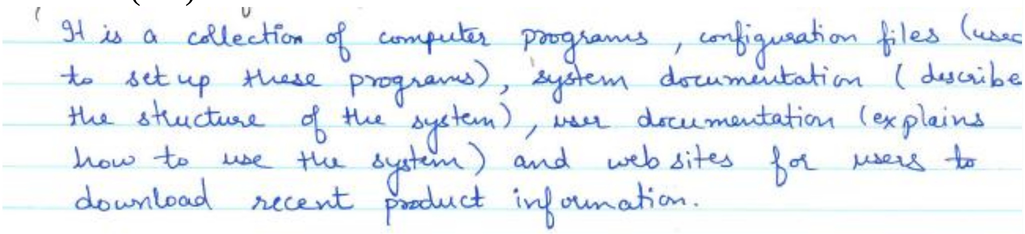
Answer FIVE FULL questions selecting AT LEAST ONE question from each part

			OBE	
		Marks	CO	RBT
1 (a)	Define software, software process model and software crisis.	[3]	CO1	L1

PART A

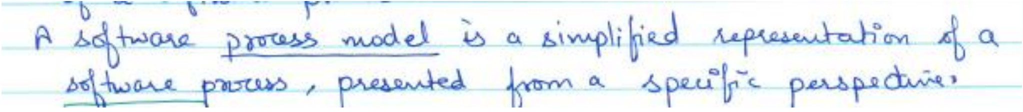
1 (a) Define software, software process model and software crisis. [3] CO1 L1

Software: (1M)



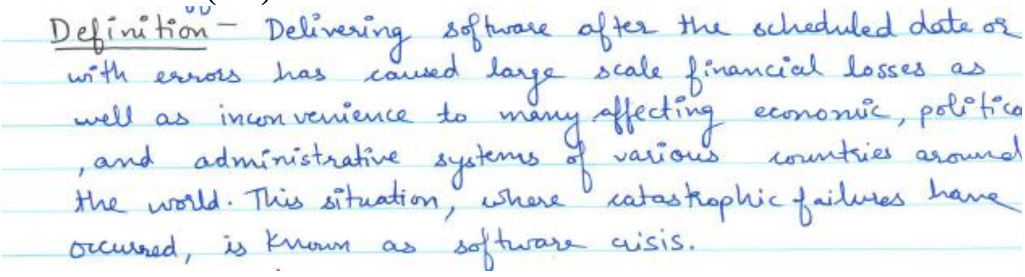
It is a collection of computer programs, configuration files (used to set up these programs), system documentation (describe the structure of the system), user documentation (explains how to use the system) and web sites for users to download recent product information.

Software process model: (1M)



A software process model is a simplified representation of a software process, presented from a specific perspective.

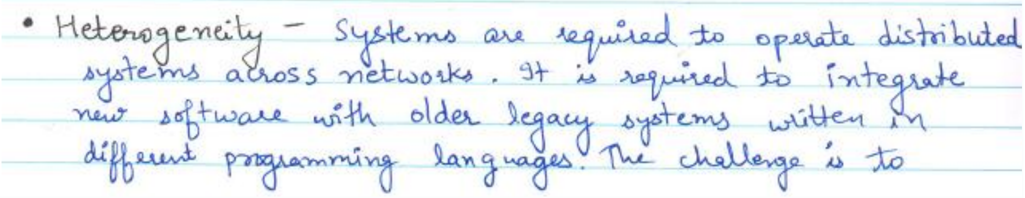
Software crisis: (1M)



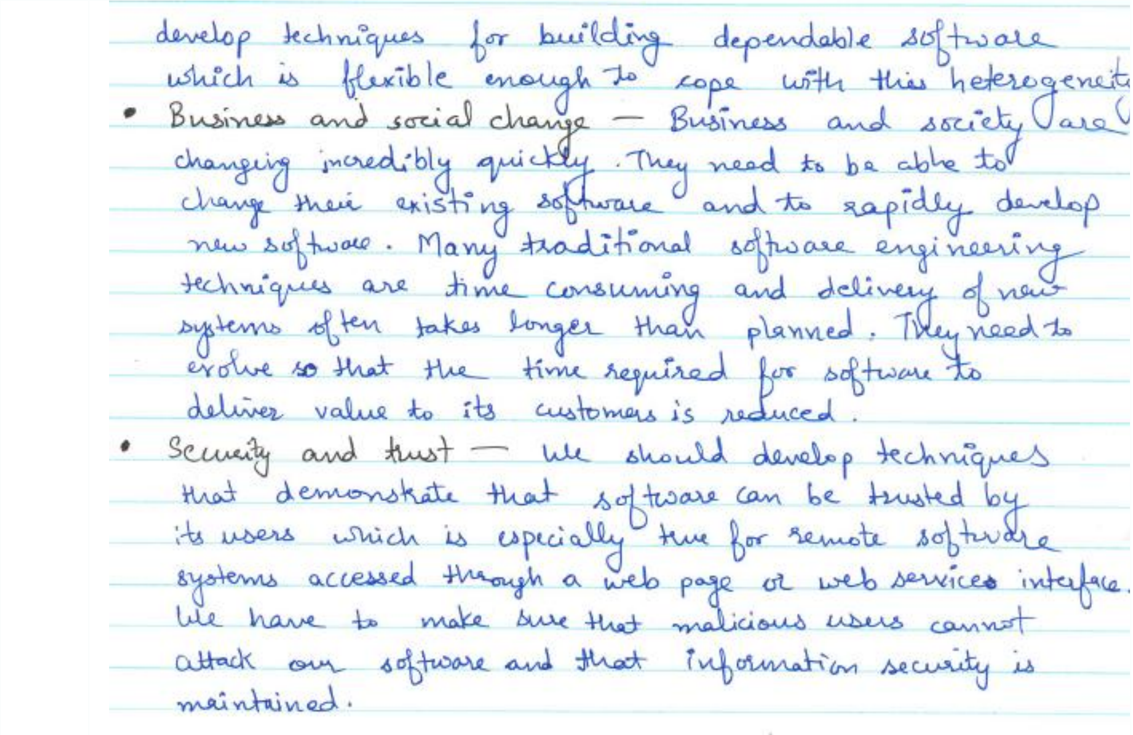
Definition - Delivering software after the scheduled date or with errors has caused large scale financial losses as well as inconvenience to many affecting economic, politica, and administrative systems of various countries around the world. This situation, where catastrophic failures have occurred, is known as software crisis.

(b) Discuss the key challenges of software engineering. [3] CO1 L2

Key challenges: (3x1M)



- Heterogeneity - Systems are required to operate distributed systems across networks. It is required to integrate new software with older legacy systems written in different programming languages. The challenge is to

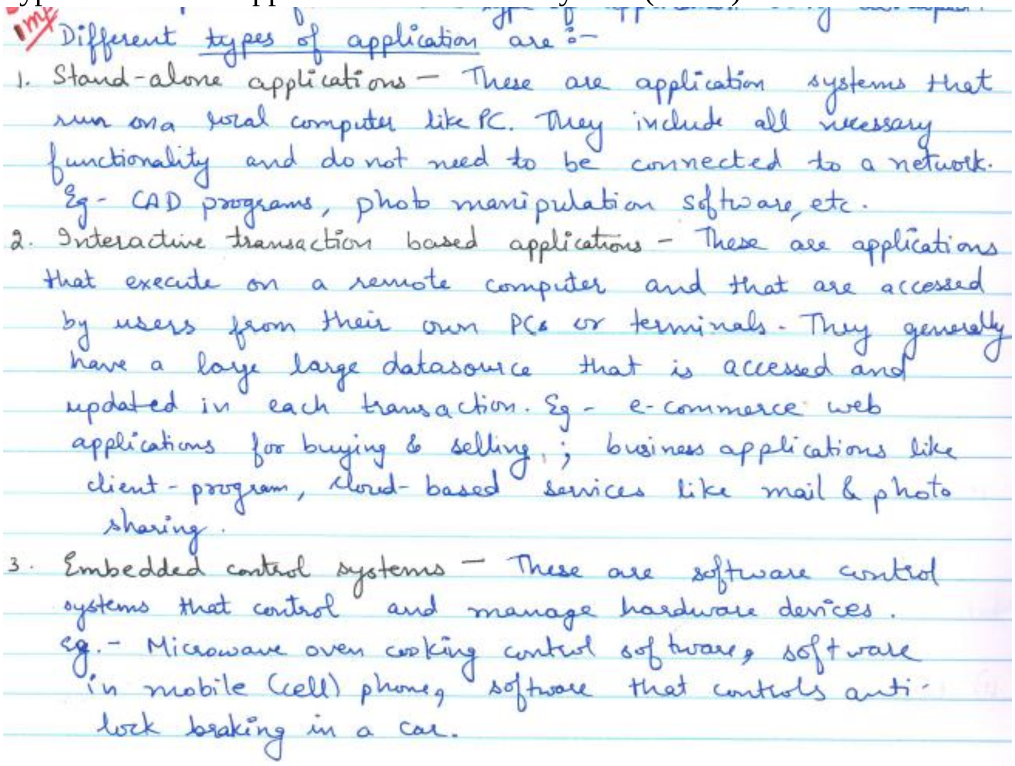


develop techniques for building dependable software which is flexible enough to cope with this heterogeneit

- Business and social change — Business and society are changing incredibly quickly. They need to be able to change their existing software and to rapidly develop new software. Many traditional software engineering techniques are time consuming and delivery of new systems often takes longer than planned. They need to evolve so that the time required for software to deliver value to its customers is reduced.
- Security and trust — We should develop techniques that demonstrate that software can be trusted by its users which is especially true for remote software systems accessed through a web page or web services interface. We have to make sure that malicious users cannot attack our software and that information security is maintained.

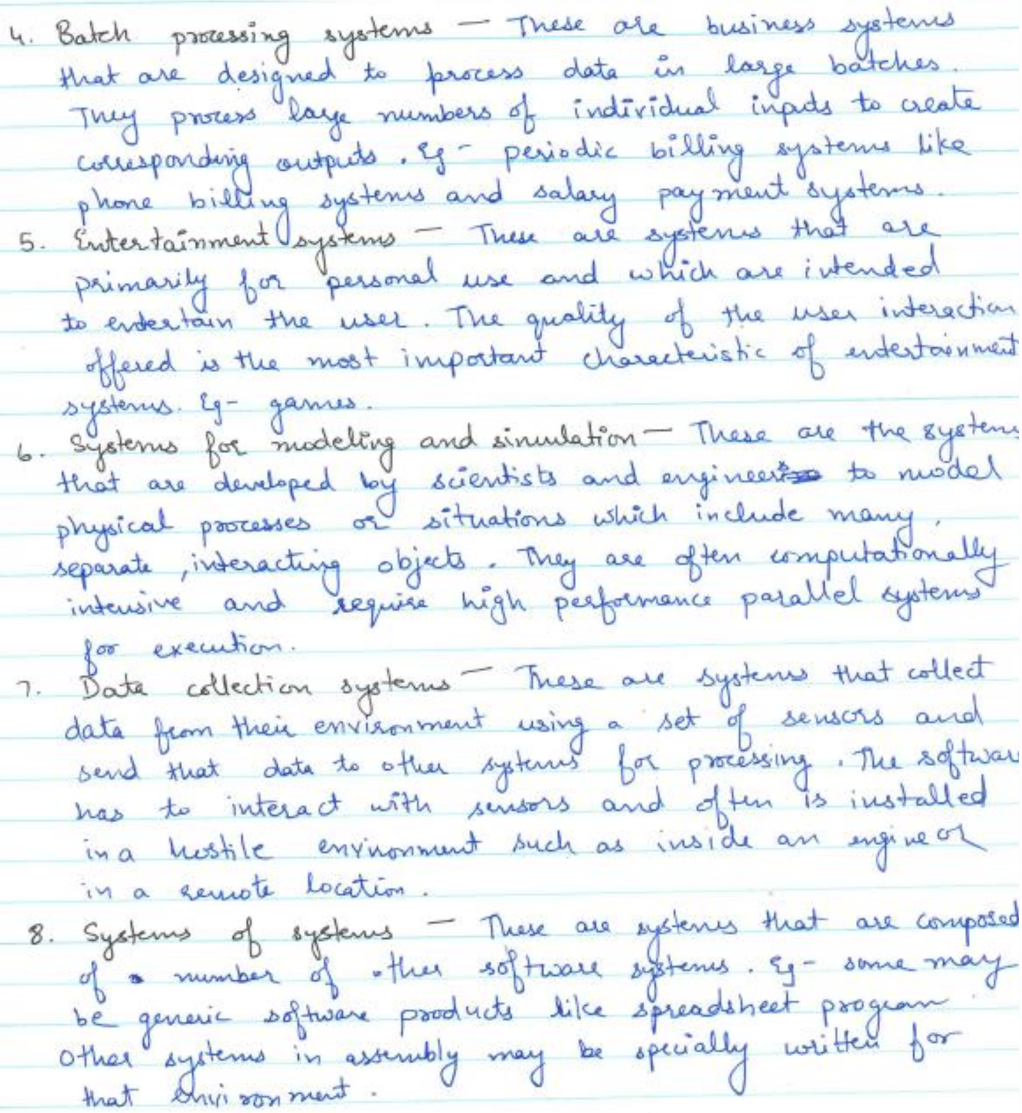
(c) Define any four types of software applications. [4] CO3 L1

Types of software applications- Mention any four (4x1M)



imp Different types of application are :-

1. Stand-alone applications — These are application systems that run on a local computer like PC. They include all necessary functionality and do not need to be connected to a network. Eg- CAD programs, photo manipulation software, etc.
2. Interactive transaction based applications - These are applications that execute on a remote computer and that are accessed by users from their own PCs or terminals. They generally have a loye large datasource that is accessed and updated in each transaction. Eg - e-commerce web applications for buying & selling, ; business applications like client-program, cloud-based services like mail & photo sharing.
3. Embedded control systems — These are software control systems that control and manage hardware devices. eg.- Microwave oven cooking control software, software in mobile (cell) phone, software that controls anti-lock braking in a car.



4. Batch processing systems — These are business systems that are designed to process data in large batches. They process laye numbers of individual inputs to create corresponding outputs. Eg- periodic billing systems like phone billing systems and salary payment systems.

5. Entertainment systems — These are systems that are primarily for personal use and which are intended to entertain the user. The quality of the user interaction offered is the most important characteristic of entertainment systems. Eg- games.

6. Systems for modeling and simulation — These are the systems that are developed by scientists and engineers to model physical processes or situations which include many, separate, interacting objects. They are often computationally intensive and require high performance parallel systems for execution.

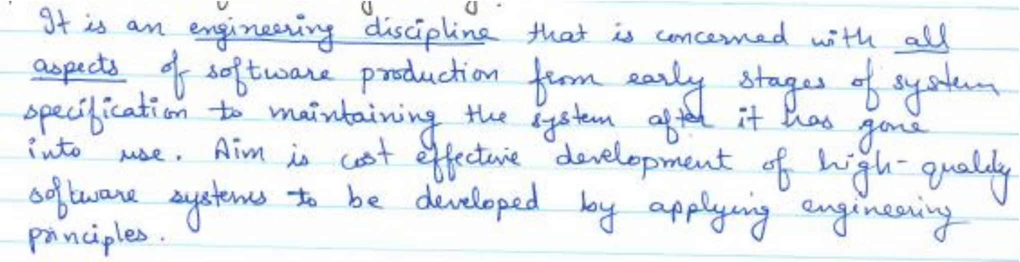
7. Data collection systems — These are systems that collect data from their environment using a set of sensors and send that data to other systems for processing. The softwar has to interact with sensors and often is installed in a hostile environment such as inside an engine or in a remote location.

8. Systems of systems — These are systems that are composed of a number of other software systems. Eg- some may be generic software products like spreadsheet program other systems in assembly may be specially written for that environment.

OR

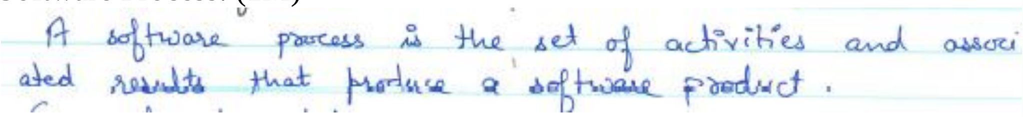
2 (a) Define software engineering, software process and professional software development. [3] CO1 L1

Software Engineering: (1M)



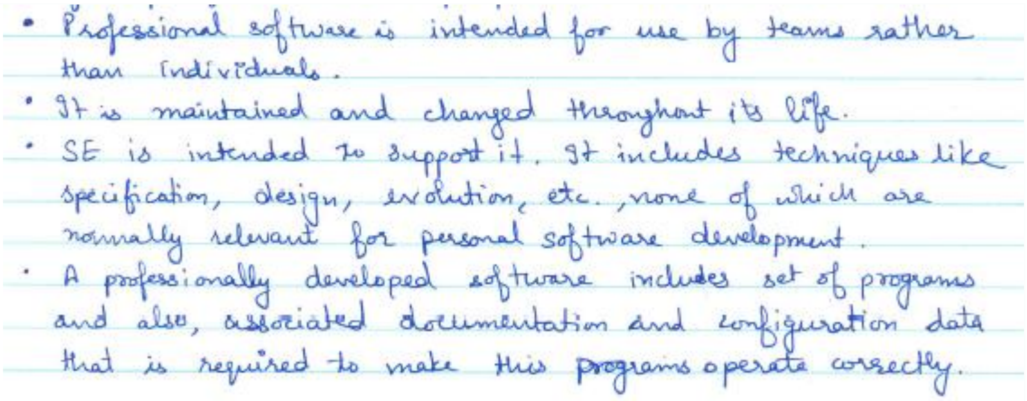
It is an engineering discipline that is concerned with all aspects of software production from early stages of system specification to maintaining the system after it has gone into use. Aim is cost effective development of high-quality software systems to be developed by applying engineering principles.

Software Process: (1M)



A software process is the set of activities and associated results that produce a software product.

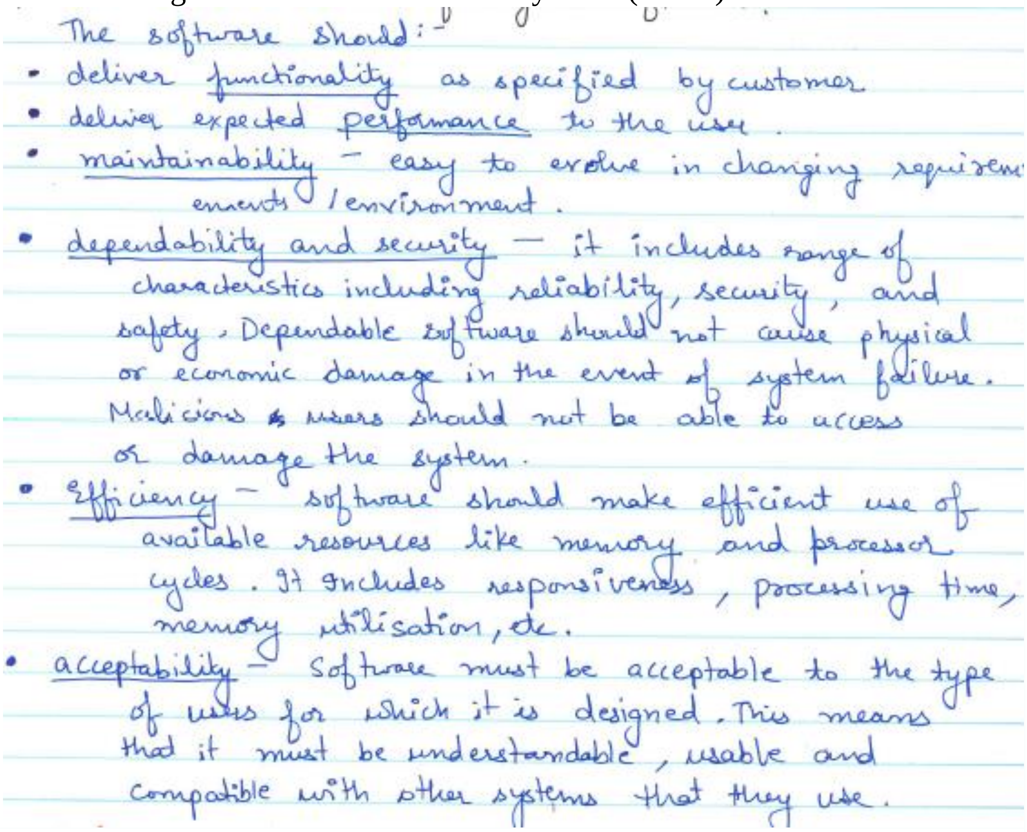
Professional software development: (1M)



- Professional software is intended for use by teams rather than individuals.
- It is maintained and changed throughout its life.
- SE is intended to support it. It includes techniques like specification, design, evolution, etc., none of which are normally relevant for personal software development.
- A professionally developed software includes set of programs and also, associated documentation and configuration data that is required to make this programs operate correctly.

(b) List and explain any three attributes of good software. [3] CO1 L2

Attributes of good software – Mention any three (3x1M)

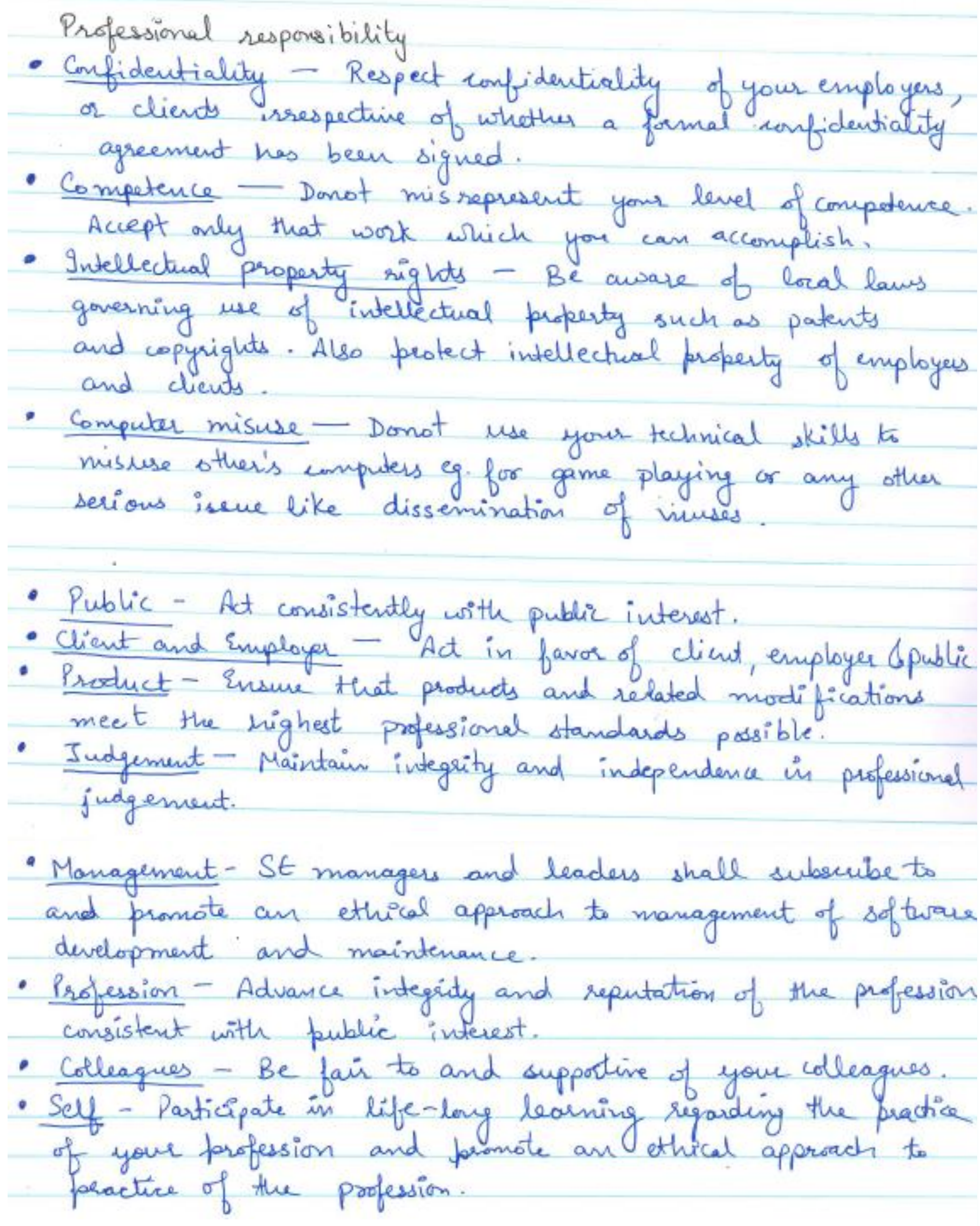


The software should:-

- deliver functionality as specified by customer
- deliver expected performance to the user.
- maintainability – easy to evolve in changing requirements / environment.
- dependability and security – it includes range of characteristics including reliability, security, and safety. Dependable software should not cause physical or economic damage in the event of system failure. Malicious users should not be able to access or damage the system.
- Efficiency – software should make efficient use of available resources like memory and processor cycles. It includes responsiveness, processing time, memory utilisation, etc.
- acceptability – Software must be acceptable to the type of users for which it is designed. This means that it must be understandable, usable and compatible with other systems that they use.

(c) Explain any four professional responsibilities. [4] CO2 L1

Professional responsibilities- Mention any four (4x1M)



Professional responsibility

- Confidentiality — Respect confidentiality of your employers, or clients irrespective of whether a formal confidentiality agreement has been signed.
- Competence — Donot misrepresent your level of competence. Accept only that work which you can accomplish.
- Intellectual property rights — Be aware of local laws governing use of intellectual property such as patents and copyrights. Also protect intellectual property of employers and clients.
- Computer misuse — Donot use your technical skills to misuse other's computers eg. for game playing or any other serious issue like dissemination of viruses.

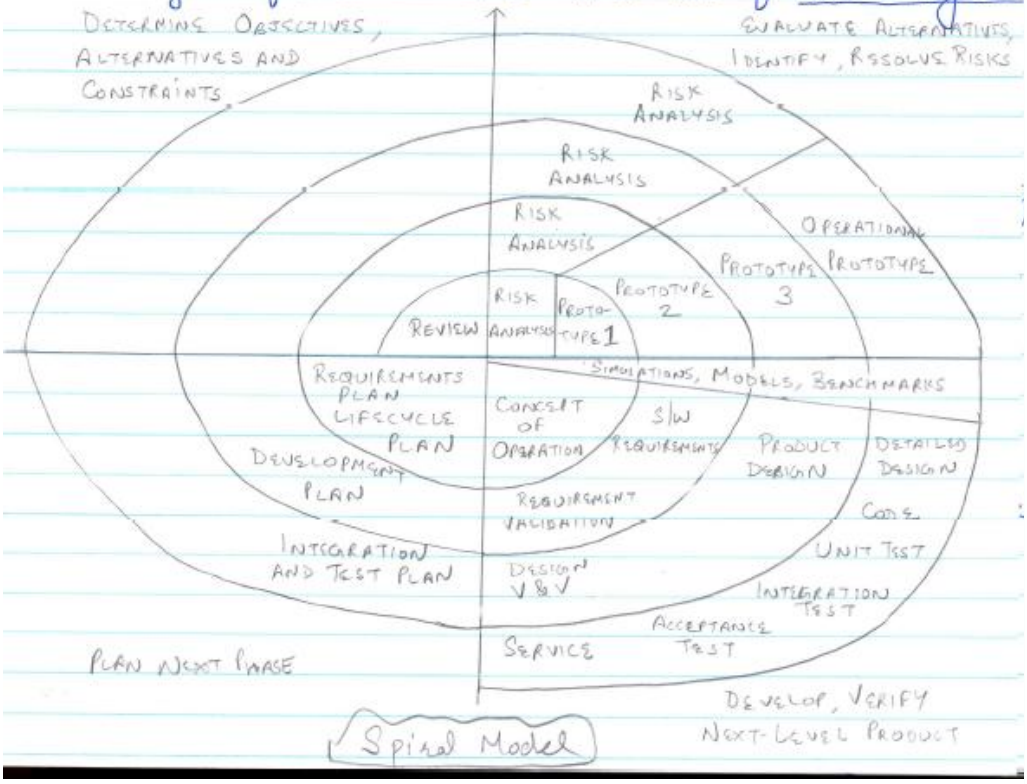
- Public - Act consistently with public interest.
- Client and Employer — Act in favor of client, employer & public
- Product — Ensure that products and related modifications meet the highest professional standards possible.
- Judgement — Maintain integrity and independence in professional judgement.

- Management - SE managers and leaders shall subscribe to and promote an ethical approach to management of software development and maintenance.
- Profession — Advance integrity and reputation of the profession consistent with public interest.
- Colleagues — Be fair to and supportive of your colleagues.
- Self - Participate in life-long learning regarding the practice of your profession and promote an ethical approach to practice of the profession.

PART B

3	Explain with a neat diagram the process model best suited for risk analysis. Spiral model (Diagram: 8M Explanation: 2M)	[10]	CO1	L2

It was proposed by Boehm in 1988. In this, the process is represented as a spiral. Each loop in the spiral represents a phase of the software process. Most important advantage of this model is involvement of risk management



Spiral Model

Each loop in the spiral is split into four sectors:

(i) Objective setting — Specific objectives, constraints, project risks for that phase of project are defined. Alternative strategies depending on these risks may be planned.

(ii) Risk assessment and reduction — For each of the identified project risks, a detailed analysis is carried out and steps are taken to reduce the risk. Risks cause schedule and cast overruns.

(iii) Development and validation — After risk evaluation, a development model for the system is chosen.

(iv) Planning — The project is reviewed and a decision is made whether to continue with a further loop of the spiral. If it is decided to continue, plans ~~of~~ are drawn up for the next phase of the project.

OR

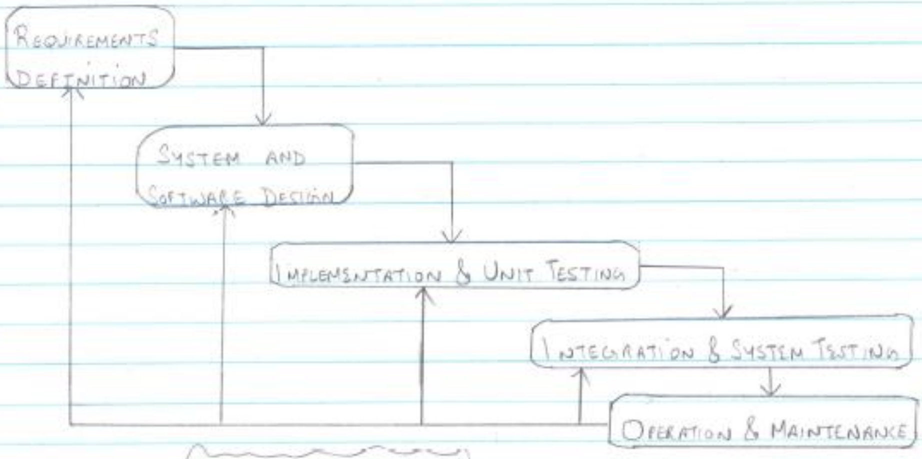
4 (a) Name the process model that has reduced cost of accommodating change, involves feedback from customer and releases software in versions. [2] CO1 L2

Incremental development (2M)

(b) Explain the steps in a waterfall model with a neat diagram. Discuss its advantages and disadvantages. [4+2+2] CO1 L2

(Waterfall model diagram: 4M, Advantages: 2M, Disadvantages: 2M)

V.imp WATERFALL MODEL


REQUIREMENTS DEFINITION
SYSTEM AND SOFTWARE DESIGN
IMPLEMENTATION & UNIT TESTING
INTEGRATION & SYSTEM TESTING
OPERATION & MAINTENANCE

The Waterfall Model

Advantages

- Documentation
- Sequential flow easy to understand

Disadvantages

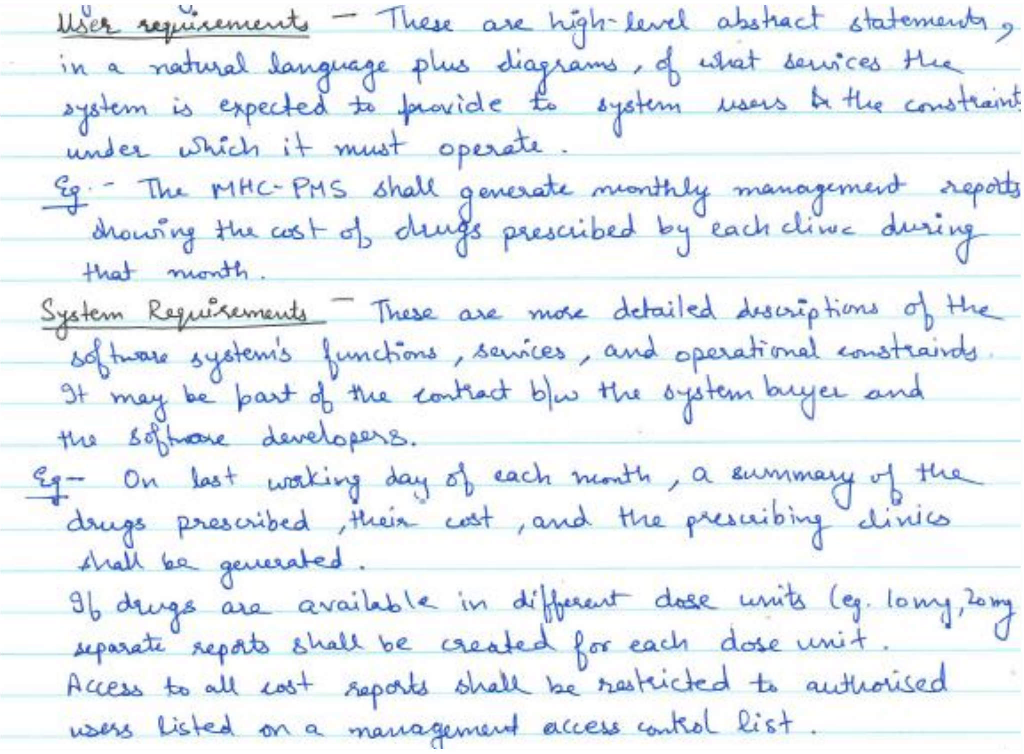
- Costs of producing & approving documents is high so iterations are costly and involve significant rework.
- Premature freezing may lead to badly structured systems.
- Partioning of project into distinct stages is inflexible
- Difficult to respond to changing customer requirements.

Use

PART C

5 (a) Differentiate between user and system requirements. Give example. [2] CO1 L2

User requirements and system requirements (Definitions: 2x0.5M, Examples: 2x0.5M)



User requirements – These are high-level abstract statements, in a natural language plus diagrams, of what services the system is expected to provide to system users & the constraint under which it must operate.

Eg. - The MHC-PMS shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

System Requirements – These are more detailed descriptions of the software system's functions, services, and operational constraints. It may be part of the contract b/w the system buyer and the software developers.

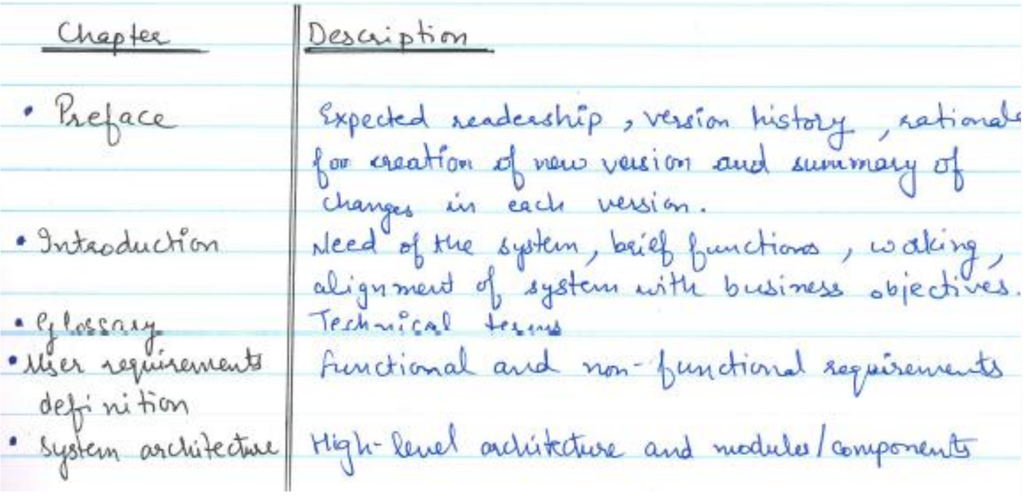
Eg- On last working day of each month, a summary of the drugs prescribed, their cost, and the prescribing clinics shall be generated.

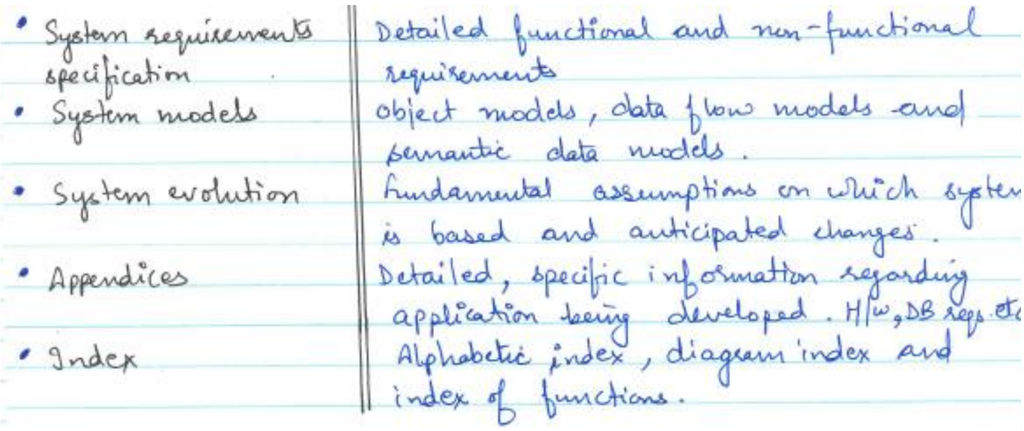
If drugs are available in different dose units (eg. 10mg, 20mg separate reports shall be created for each dose unit.

Access to all cost reports shall be restricted to authorised users listed on a management access control list.

(b) Explain the structure of SRS. [8] CO4 L2

Structure of SRS: (8M)





Chapter	Description
• Preface	Expected readership, version history, rationale for creation of new version and summary of changes in each version.
• Introduction	Need of the system, brief functions, working, alignment of system with business objectives.
• Glossary	Technical terms
• User requirements definition	functional and non-functional requirements
• System architecture	High-level architecture and modules/components
• System requirements specification	Detailed functional and non-functional requirements
• System models	object models, data flow models and semantic data models.
• System evolution	fundamental assumptions on which system is based and anticipated changes.
• Appendices	Detailed, specific information regarding application being developed. H/w, DB reqs etc
• Index	Alphabetic index, diagram index and index of functions.

OR

6 (a) Differentiate between functional and non-functional requirements. Give example. [2] CO1 L2

Functional requirements and non-functional requirements (Definitions: 2x0.5M, Examples: 2x0.5M)

FUNCTIONAL REQUIREMENTS

These describe what the system should do. These requirements depend on the type of software being developed, the expected users of the software, and the general approach taken by organization when writing requirements.

Eg :- functional requirements for MHC-PMS :-

- A user shall be able to search the appointments lists for all clinics.
- The system shall display each day, for each clinic, a list of patients who are expected to attend appointments that day.
- Each staff member using the system shall be uniquely identfie by his/her eight digit employee number.

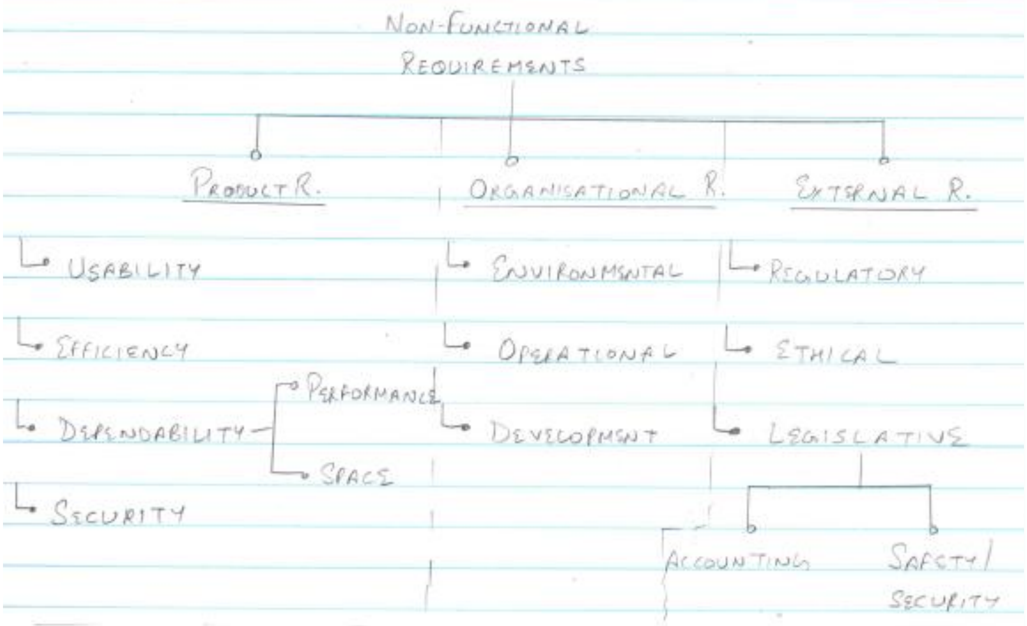
Functional requirements specification should be both complete (all services required by the user should be defined) and consistent (requirements should not have contradictory definitions.

NON-FUNCTIONAL REQUIREMENTS

They relate to emergent system properties like reliability, response time etc and may be more critical than individual functional requirements. Failing to meet a non-functional requirement can mean that whole system is unsuable.

(b) Explain the types of non-functional requirements with examples. [8] CO1 L2

Types of non-functional requirements: 5M, Examples: 3x1M)



- Product requirements – Specify or constrain behaviour of the software.
- Organizational requirements – These are derived from policies and procedures in the customer's and developer's organization.
- External requirements – These are derived from the factors external to the system and its development process.

Eg.

PRODUCT REQUIREMENT –

The MHC-PMS shall be available to all clinics during normal working hours (Mon-Fri, 08:30-17:30). Downtime within normal working hours shall not exceed five seconds in any one day.

ORGANIZATIONAL REQUIREMENT –

Users of MHC-PMS system shall authenticate themselves using their health authority identity card.

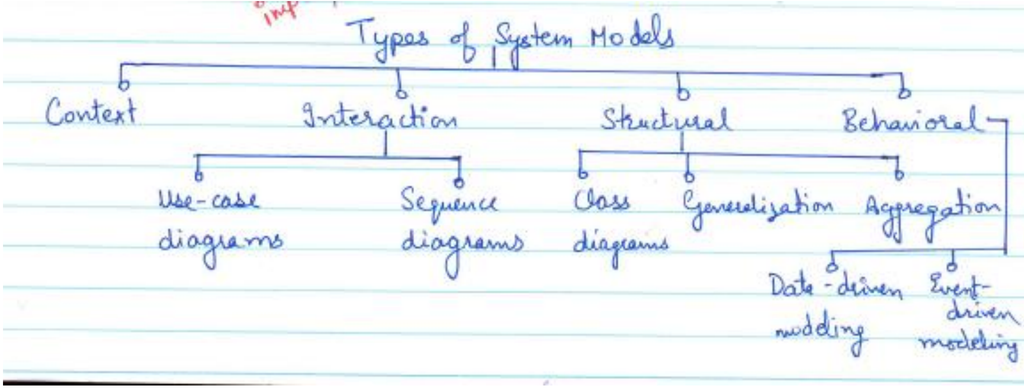
EXTERNAL REQUIREMENT –

The system shall implement patient privacy provisions as set out in HStan-03-2006-priv.

PART D

7 (a) Explain the types of system models. [7] CO5 L2

Types of system models (Names: 4M, Definitions: 3M)

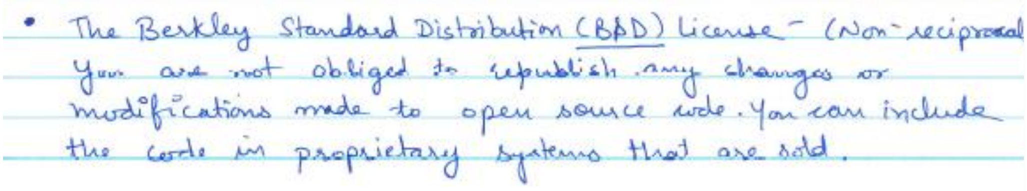


(b) Define three license models for open source licensing. [3] CO5 L2

Open source license models: (3x1M)

- The GNU General Public License (GPL) – (Reciprocal license)
  If you use open source software that is licensed under the GPL license, then you must make that software open source

The GNU Lesser General Public License (LGPL) –
It is variant of GPL which allows you to write components that link to open source code without having to publish the source of those components.

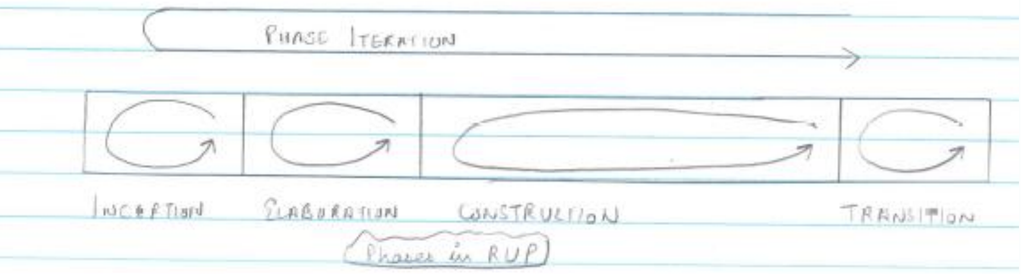


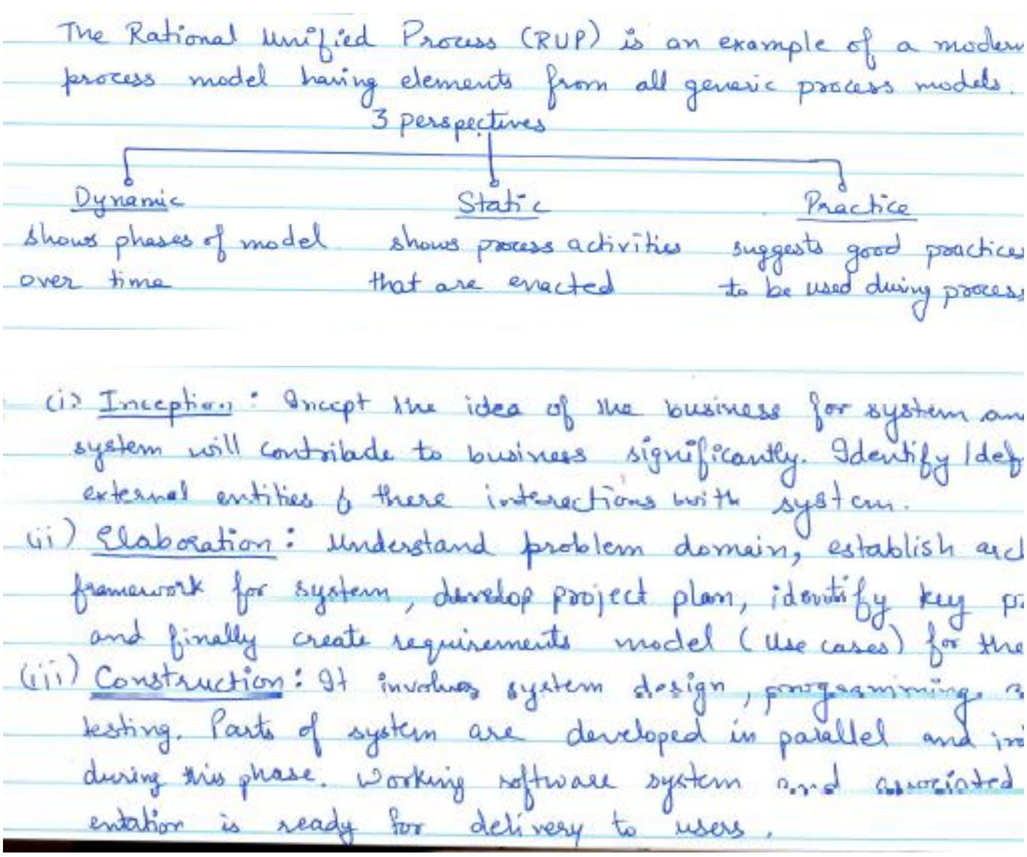
- The Berkley Standard Distribution (BSD) License - (Non-reciprocal
You are not obliged to republish any changes or modifications made to open source code. You can include the code in proprietary systems that are sold.

OR

8 (a) Explain with a neat diagram rational unified process. [5] CO1 L2

Rational Unified Process: (Diagram: 3M, Explanation: 2M)





The Rational Unified Process (RUP) is an example of a modern process model having elements from all generic process models.

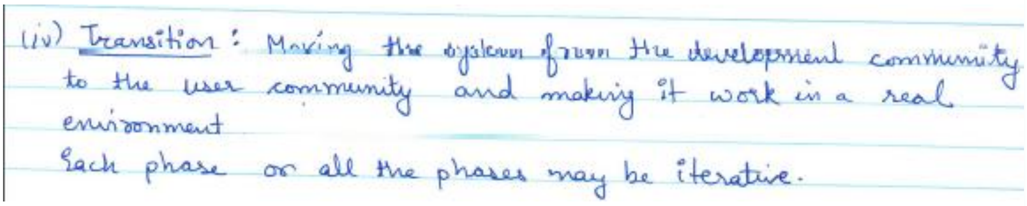
3 perspectives

Dynamic	Static	Practice
shows phases of model over time	shows process activities that are enacted	suggests good practices to be used during process

(i) Inception: Incept the idea of the business for system an system will contribute to business significantly. Identify Idef external entities & there interactions with system.

(ii) Elaboration: Understand problem domain, establish arc framework for system, develop project plan, identify key p and finally create requirements model (Use cases) for the

(iii) Construction: It involves system design, programming, a testing. Parts of system are developed in parallel and in during this phase. Working software system and associated entation is ready for delivery to users.



(iv) Transition: Moving the system from the development community to the user community and making it work in a real environment

Each phase or all the phases may be iterative.

(b) Explain design patterns with example. [5] CO4 L2

Design Patterns: (Example: 3M, Explanation: 2M)

DESIGN PATTERNS

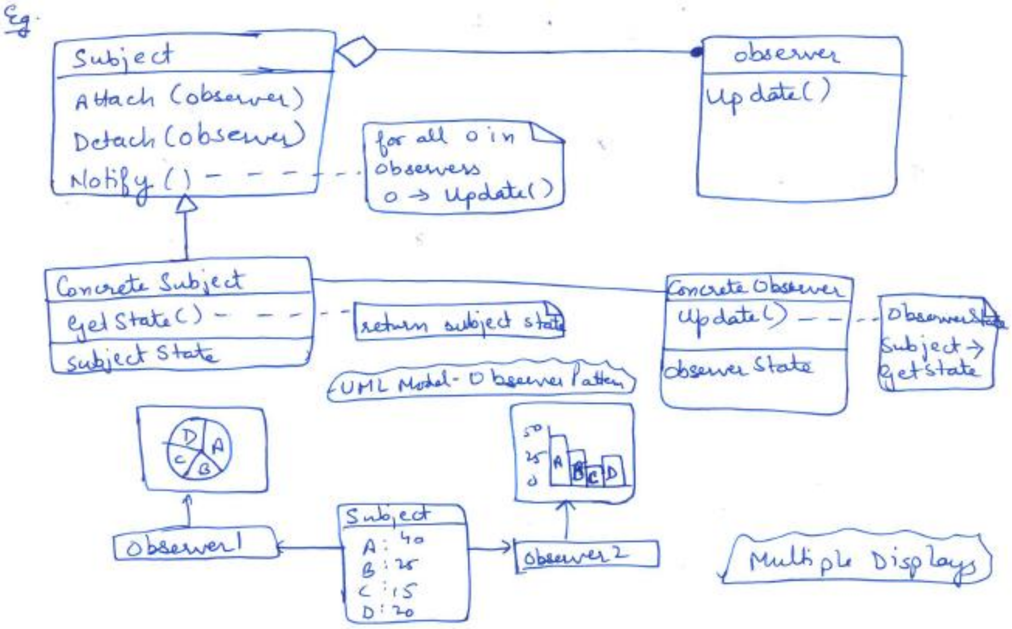
It is a general reusable solution to a commonly occurring problem, within a given context in software design. It is not a finished design that can be transformed directly into source or machine code. It is a description or template for how to solve a problem that can be used in many different situations. It allows reusing abstract knowledge about a problem & its solution.

Essential elements of design patterns:

- Name - pattern identifier
- Problem description - when to apply
- Solution description - template of design solution that can be instantiate in dif. ways.

Consequences. - results & trade offs.

Eg.



Subject: Interface/Abstract class defining ops for attaching & detaching observers to the client.

Concrete Subject: Maintains the state of object & when a change in the state occurs, it notifies attached observers.

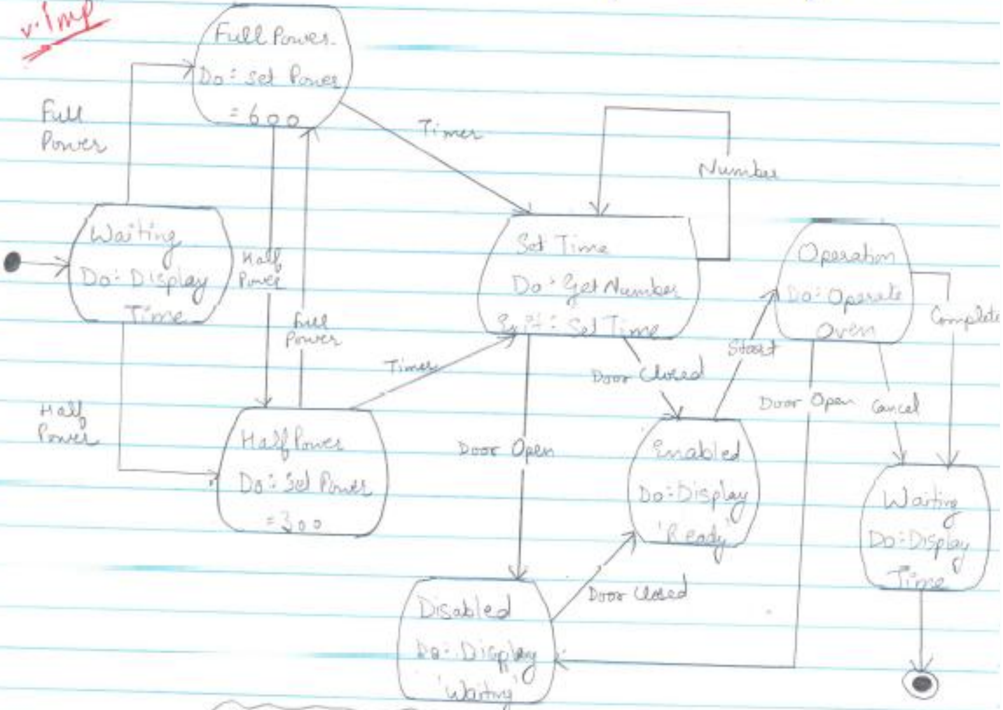
Observer: interface or abstract class defining the ops to be used to notify this object

Concrete object - concrete Observer implementations.

PART E

9 Draw and explain the state diagram of microwave oven. [10] CO5 L3

State diagram of microwave oven (Diagram: 8M, Explanation: 2M) v.Imp

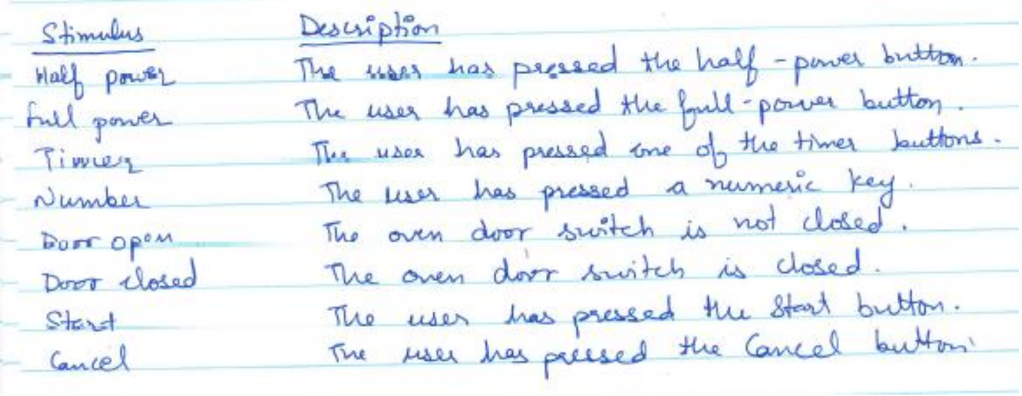


State diagram of a microwave oven

Sequence of actions assumed:

- Select the power level (either half power or full power)
- Input the cooking time using a numeric keypad.
- Press Start and the food is cooked for the given time.

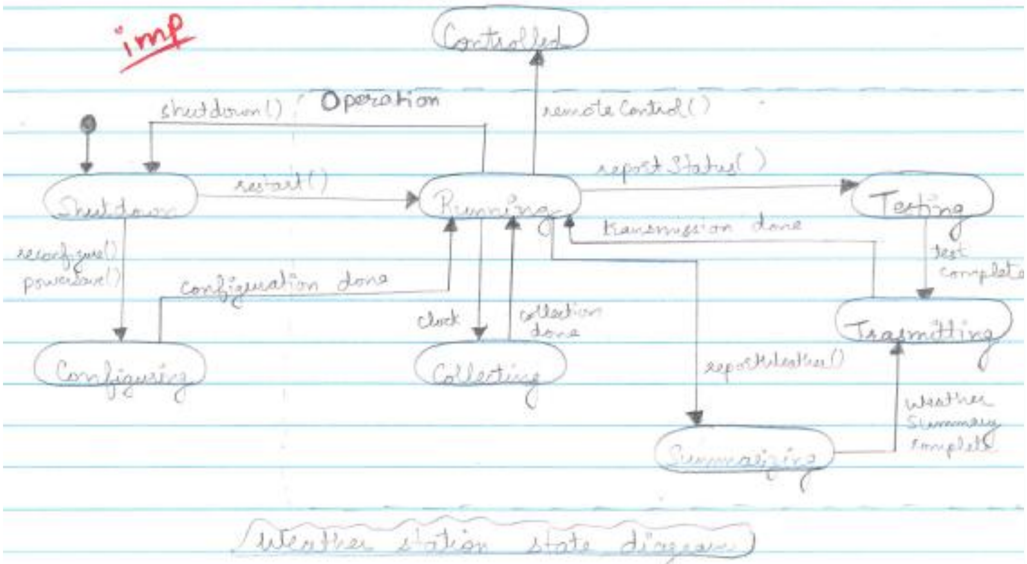
State	Description
Waiting	Oven is waiting for input. Display shows current time.
Half power	Oven power is set to 300 watts. Display shows 'Half power'.
Full power	Oven power is set to 600 watts. Display shows 'Full power'.
Set time	Cooking time is set to the user's input value. The display shows the cooking time selected and is updated as the time is set.
Disabled	Oven operation is disabled for safety. Interior oven light is on. Display shows 'Not ready'.
Enabled	Oven operation is enabled. Interior oven light is off. Display shows 'Ready to cook'.
Operation	Oven in operation. Interior oven light is on. Display shows the timer countdown. On completion of cooking, the buzzer is sounded for five seconds. Oven light is on. Display shows 'Cooking complete' while buzzer is sounding.



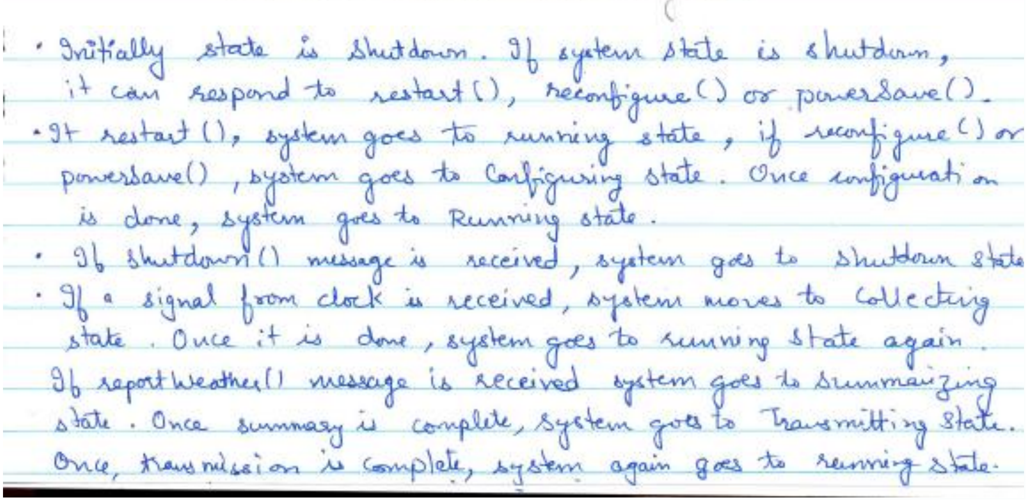
Stimulus	Description
Half power	The user has pressed the half-power button.
Full power	The user has pressed the full-power button.
Timer	The user has pressed one of the timer buttons.
Number	The user has pressed a numeric key.
Door open	The oven door switch is not closed.
Door closed	The oven door switch is closed.
Start	The user has pressed the Start button.
Cancel	The user has pressed the Cancel button.

OR

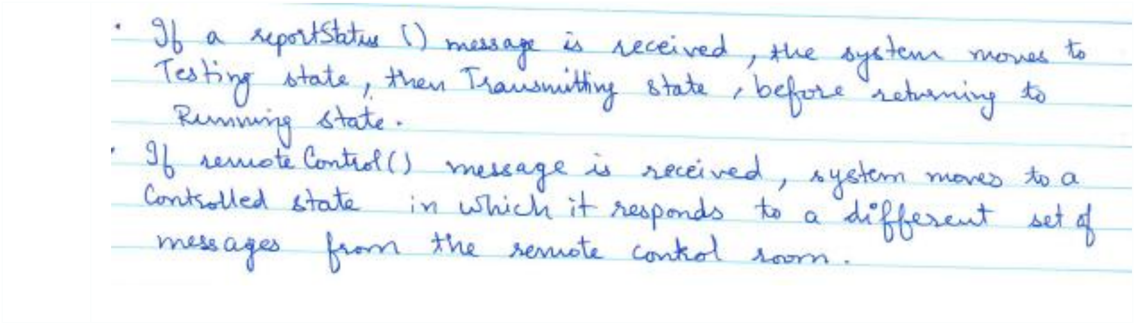
10	Draw and explain the state diagram of weather station system. State diagram of weather station system (Diagram: 8M, Explanation: 2M)	[10]	CO5	L3



Weather station state diagram



- Initially state is Shutdown. If system state is shutdown, it can respond to restart(), reconfigure() or powerSave().
- If restart(), system goes to running state, if reconfigure() or powersave(), system goes to Configuring state. Once configuration is done, system goes to Running state.
- If shutdown() message is received, system goes to shutdown state
- If a signal from clock is received, system moves to Collecting state. Once it is done, system goes to running state again. If reportWeather() message is received system goes to summarizing state. Once summary is complete, system goes to Transmitting state. Once, transmission is complete, system again goes to running state.



- If a reportStatus () message is received, the system moves to Testing state, then Transmitting state, before returning to Running state.
- If remoteControl() message is received, system moves to a Controlled state in which it responds to a different set of messages from the remote control room.

Course Outcomes		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1:	Design a software system, component, or process to meet desired needs within realistic constraints	3	3	3	3	2	3	2	-	2	2	2	3
CO2:	Assess professional and ethical responsibility	2	-	-	-	-	3	2	3	3	2	-	3
CO3:	Function on multi-disciplinary teams	2	2	3	3	2	2	2	-	3	3	2	3
CO4:	Use the techniques, skills, and modern engineering tools necessary for engineering practice	3	3	3	3	3	2	2	-	2	2	2	3
CO5:	Analyze, design, implement, verify, validate, implement, apply, and maintain software systems or parts of software systems	3	3	3	3	2	2	2	-	2	2	2	3

Cognitive level	KEYWORDS
L1	List, define, tell, describe, identify, show, label, collect, examine, tabulate, quote, name, who, when, where, etc.
L2	summarize, describe, interpret, contrast, predict, associate, distinguish, estimate, differentiate, discuss, extend
L3	Apply, demonstrate, calculate, complete, illustrate, show, solve, examine, modify, relate, change, classify, experiment, discover.
L4	Analyze, separate, order, explain, connect, classify, arrange, divide, compare, select, explain, infer.
L5	Assess, decide, rank, grade, test, measure, recommend, convince, select, judge, explain, discriminate, support, conclude, compare, summarize.

PO1 - *Engineering knowledge*; PO2 - *Problem analysis*; PO3 - *Design/development of solutions*; PO4 - *Conduct investigations of complex problems*; PO5 - *Modern tool usage*; PO6 - *The Engineer and society*; PO7- *Environment and sustainability*; PO8 – *Ethics*; PO9 - *Individual and team work*; PO10 - *Communication*; PO11 - *Project management and finance*; PO12 - *Life-long learning*