Subcribe my youtube channel Habiba's Lectures for video lectures

Professional Practices Complete notes by Habiba

For IT/CS

Subcribe for video lectures by clicking on this logo

https://www.youtube.com/channel/UCprAYzuA8vAPPJP3HCIG2FA

Consequences of Not Following Ethical Codes in Computing

Ethical codes in computing provide guidelines for the responsible and fair use of technology. These codes help professionals make decisions that protect users, data, and systems. Failing to follow these ethical standards can lead to serious consequences for individuals, organizations, and society.

1. Legal Consequences

If an IT professional engages in unethical activities—such as:

- Data breaches
- Identity theft
- Unauthorized hacking
- Software piracy

They may face legal actions, including:

- Fines
- Jail time
- Court cases
- Cybercrime charges

Example: In Pakistan, cybercrimes are punishable under the **PECA Act**.

2. Loss of Trust

If a company or developer acts unethically—such as:

- Leaking customer data
- Launching fake products
- Creating biased algorithms

People lose trust in the brand.

Result: Customers stop buying products, and the business reputation gets damaged.

3. Job Loss or Career Damage

Companies expect IT professionals to act responsibly. If someone commits unethical acts like:

- Unauthorized data access
- Code manipulation
- False documentation
- Misusing company tools

They can be immediately fired, and their future job opportunities may be ruined.

4. Security Breaches

Unethical practices can lead to:

- Virus attacks
- Hackers gaining system access
- Insecure software

Example: A careless developer using weak password encryption can lead to a full system hack.

5. Financial Loss

If unethical actions occur—such as:

- Intentional software bugs
- Data leaks
- Unauthorized tool usage

The company may face:

- Lawsuits
- Loss of customers
- Decreased revenue

Sometimes, the financial damage can be in **billions**.

6. Harm to Users

Unethical computing can **directly harm people** in real life, such as:

- Bank account hacking
- Medical record leaks
- Online bullying

This causes **psychological, emotional, and financial damage** to victims.

7. Violation of Privacy

Ethical computing means **respecting user privacy**. If personal data is shared without permission, it violates privacy rights.

Example: Facebook has been fined **billions** for privacy violations.

8. Bad Reputation for the IT Industry

If too many IT professionals act unethically, the **entire industry's reputation suffers**. People start thinking:

- "IT professionals can't be trusted."
- "All software is risky."
- "No one's data is safe."

This fear discourages new talent from entering the field.

So, Following ethical codes in computing is **essential** to maintain trust, security, and fairness in technology. Ignoring them leads to **legal trouble, financial loss, and harm to society**. Responsible computing ensures a **safer and more reliable digital world** for everyone.

How Can Computing Professionals Protect the Public Interest?

Introduction:
Public interest means protecting the rights and well-being of ordinary people. Computing professionals (such as software engineers, developers, and IT experts) don't just write code—they have a responsibility to ensure that the systems, apps, and technologies they create are **safe, fair, and beneficial** for society. If they make decisions based only on company profits or personal gain, the public can suffer harm.

1. Ensuring Privacy and Data Protection

The most important step in protecting public interest is **safeguarding users' personal data**.

- Professionals should build systems with **strong encryption, secure logins, and access controls**.
- Sharing or selling user data **without consent** is unethical.
- **Example:** Social media apps should not share users' private information with third parties without permission.

2. Developing Secure Systems

Software with **bugs, loopholes, or weak security** makes people vulnerable to hackers.

- Professionals must follow **secure coding practices**.
- Regular **updates and security patches** should be provided to protect against threats.

3. Avoiding Harmful Content or Bias

AI, search engines, and recommendation systems can **promote unfairness** if they are biased.

- Professionals must design **fair algorithms** and moderate content responsibly.
- No app or system should **favor or discriminate** based on race, gender, or religion.

4. Designing for Accessibility

Public interest means **everyone should benefit from technology**, including people with disabilities.

- Systems should support **screen readers, high-contrast modes, and keyboard navigation**.
- Mobile apps and websites must be **inclusive and user-friendly for all**.

5. Reporting and Preventing Misuse

If a professional sees their software being used for **illegal or unethical purposes**, they must **speak up**.

- **Example:** If a surveillance tool is being misused, the developer has a duty to **oppose it**.

- Staying silent is **also unethical**.

6. Environmental Responsibility

Tech products should be designed with **sustainability** in mind.

- **Energy-efficient** software and hardware reduce carbon footprints.
- Data centers and cloud services should **minimize environmental impact**.

7. Promoting Digital Literacy

Professionals can help the public by:

- Hosting **seminars, workshops, and awareness campaigns**.
- Teaching people how to **protect their data, avoid scams, and use privacy settings**.

8. Following Professional Ethical Codes

Organizations like **ACM and IEEE** have established **ethical guidelines** for computing professionals.

- Following these codes ensures that decisions **benefit the public** rather than just corporations.

So, Computing professionals have a **duty to society**. By prioritizing **privacy, security, fairness, and accessibility**, they can ensure technology serves the **greater good**—not just profits. Ethical choices today lead to a **safer and fairer digital future** for everyone.

Comparison of Computing Profession with Medicine and Law

Introduction

A profession is a field that requires specialized knowledge, training, and a commitment to public service. Medicine, law, and computing are all respected professions, but each has unique characteristics. This comparison will help explain why computing is also considered a responsible profession.

1. Formal Education & Specialized Knowledge

Profession	Requirements
Medicine	Doctors need an MBBS degree, specialization (e.g., cardiology), and clinical training.
Law	Lawyers must complete an LLB degree and pass the bar council exam. They must have in-depth knowledge of laws and court procedures.
Computing	Computing professionals earn degrees like BCS, BSSE, or BSIT. They require expertise in programming, AI, cybersecurity, and system design.

Conclusion: All three professions require **specialized knowledge and formal education**.

2. Code of Ethics and Professional Conduct

Profession	Ethical Guidelines
Medicine	Doctors take the **Hippocratic Oath**—"First, do no harm."
Law	Lawyers must uphold justice, maintain client confidentiality, and follow legal ethics.
Computing	Organizations like **ACM and IEEE** define ethical codes, such as protecting user privacy and developing secure software.

Conclusion: All three fields have **strict ethical guidelines** that professionals must follow.

3. Public Interest and Trust

Profession	Public Trust
Medicine	People trust doctors with their **health and lives**.
Law	People trust lawyers with their **legal rights and justice**.
Computing	People trust software engineers with their **data, identity, and digital security**.

Conclusion: All three professions **directly impact public welfare and trust**.

4. Licensing and Certification

Profession	Licensing Requirements
Medicine	Doctors must obtain a **medical license** to practice.
Law	Lawyers must be **licensed by the bar council**.
Computing	No **mandatory licensing**, but certifications (Microsoft, Cisco, AWS) boost credibility.

Key Difference: Unlike medicine and law, computing does not yet have **compulsory licensing**, but certifications play a major role.

5. Life and Social Impact

Profession	Impact on Society
Medicine	A doctor's decision can **save or end a life**.
Law	A lawyer's case can determine **freedom or punishment**.
Computing	A software bug or biased AI can **affect millions** (e.g., privacy breaches, medical device failures, autonomous car accidents).

Conclusion: Computing has a **deep and critical impact** on society, just like medicine and law.

6. Continued Learning and Updating Knowledge

Profession	Need for Continuous Learning
Medicine	Doctors must stay updated on **new diseases and treatments**.
Law	Lawyers must track **new laws and court rulings**.
Computing	Professionals must learn **new programming languages, security threats, and frameworks**.

Conclusion: All three fields require **lifelong learning** to stay relevant.

The computing profession shares key characteristics with medicine and law:

- **Specialized education**
- **Public responsibility**
- **Ethical codes**
- **Continuous learning**

While **medicine and law** have well-established legal and social structures, computing is still a **younger profession**. However, its impact on society is **huge and high-risk**, meaning computing professionals must be **just as responsible and ethical** as doctors and lawyers.

Computing is not just about coding—it's a **serious profession** that affects lives, rights, and security. Ethical responsibility is **essential** in all three fields.

Comparison of Partnership and Company Business Structures

1. Partnership

Definition:

A partnership is a business structure where two or more individuals jointly operate a business based on mutual agreement and share profits/losses.

Legal Definition (Under Partnership Act 1932):

"A relationship between persons who have agreed to share the profits of a business carried on by all or any of them acting for all."

Key Features of Partnership:

- **Minimum 2 partners** required
- **Maximum 20 partners** allowed (for general businesses)
- Business is run based on mutual agreement (oral or written, known as "Partnership Deed")
- **Unlimited liability** – Personal assets can be used to cover business losses
- **Joint liability** – All partners are responsible for any one partner's mistakes
- **Registration is optional**, but a registered partnership is more secure

Example:

If Ali, Ahmed, and Sara start a bakery business together and share profits/losses, it is considered a partnership.

2. Company

Definition:

A company is a legal entity formed under corporate law. It has its own legal identity, meaning it can own property, sue, or be sued in its own name.

Legal Definition (Under Companies Act):

"A company is an artificial person, created by law, having separate legal identity and perpetual succession."

Key Features of a Company:

- **Separate Legal Entity** – The company is legally distinct from its shareholders
- **Limited Liability** – Shareholders are only liable up to their invested amount; personal assets are protected
- **Perpetual Succession** – The company continues to exist even if shareholders or directors change
- **Transferability of Shares** – Shares can be easily bought/sold (for public companies)
- **Compulsory Registration** – Must be registered with SECP (in Pakistan)

Types of Companies:

1. **Private Limited Company (Pvt. Ltd.)** – Small companies with limited shareholders
2. **Public Limited Company (Ltd.)** – Large companies listed on stock exchanges

3. **Single Member Company (SMC)** – Owned by one person with limited liability

Example:

Nestle Pakistan Ltd. or Unilever are companies with separate legal identities, shareholders, and boards of directors.

Major Differences Between Partnership and Company

Aspect	Partnership	Company
Legal Status	Not a separate legal entity	Separate legal entity
Liability	Unlimited (personal assets at risk)	Limited (only invested amount at risk)
Formation	Simple (agreement-based)	Complex (requires legal registration)
Members	2 to 20 partners	Minimum 1 (SMC) or more
Registration	Optional	Mandatory
Perpetual Succession	No (ends if partners leave/die)	Yes (continues indefinitely)
Transfer of Ownership	Difficult (requires new agreement)	Easy (through share transfers)

- **Partnerships** are simpler to form but have **higher financial risk** due to unlimited liability.
- **Companies** provide **legal protection** (limited liability) but require **more formalities** in setup and regulation.

Choosing between them depends on business size, risk tolerance, and long-term goals.

Organizational Structures: Market-Based vs. Technology-Based

1. Market-Based Structure (Customer-Centric Approach)

Definition:

In this structure, an organization divides its departments or teams based on different customer groups or market segments. Each unit specializes in serving a particular market.

Key Characteristics:

- Organizational structure aligns with customer needs
- Separate teams for each market segment
- Each unit develops expertise in its assigned sector

Advantages:

- Better understanding and service for customer needs
- Customized strategies for each market segment
- More focused business development and sales

Disadvantages:

- Possible duplication of resources (e.g., separate HR teams for each sector)
- Coordination challenges between departments

Example:

An IT company working in healthcare, education, and finance sectors may have:

- **Healthcare Solutions Team**
- **Education Solutions Team**
- **Finance Solutions Team**

2. Technology-Based Structure (Functional/Technical Approach)

Definition:

In this model, an organization groups teams by technical functions or expertise areas, such as programming, networking, UI/UX design, and testing.

Key Characteristics:

- Structure is based on technical skills and specialization
- Each department handles a specific technical function

- Projects require collaboration between cross-functional teams

Advantages:

- Strong technical expertise and skill development
- Easier supervision due to grouped specializations
- Efficient use of resources

Disadvantages:

- Less focus on customer needs
- Requires strong inter-departmental coordination
- Teams may prioritize technical aspects over customer solutions

Example:

A software company may structure its teams as:

- **Front-End Development Team**
- **Back-End Development Team**
- **Database Team**
- **QA/Testing Team**
- **DevOps Team**

Comparison Table: Market-Based vs. Technology-Based Structures

Feature	Market-Based Structure	Technology-Based Structure
Basis of Division	Customers / Market Segments	Technical Skills / Functions
Primary Focus	Customer needs & market adaptation	Technical excellence & efficiency
Flexibility	High (adapts to market demands)	High (optimized for development)
Communication Flow	Within market units	Within technical departments
Resource Utilization	Possible duplication	Shared efficiently
Example Divisions	Healthcare, Education, Retail	Programming, Design, Testing

- **Market-based structures** are ideal for customer-focused businesses needing tailored solutions.
- **Technology-based structures** work best for technical organizations prioritizing skill specialization.
- The choice depends on whether the company prioritizes **customer relationships** or **technical efficiency**.

Fixed Assets vs. Current Assets:

Fixed Assets (Non-Current Assets)

Definition:

Fixed assets are **long-term tangible assets** used in business operations for **more than one year**. These assets are not meant for sale but rather for **sustaining business activities**.

Key Characteristics:

- **Tangible** (physical, can be seen and touched)
- **Long-term use** (more than 1 year)
- **Depreciate over time** (lose value gradually)

Examples:

- Buildings (office, factory)
- Machinery (production equipment)
- Vehicles (company cars, trucks)
- Furniture (office desks, chairs)

Example Sentence:

"A building used for office operations is a fixed asset."

Current Assets

Definition:

Current assets are **short-term assets** that can be **converted into cash within one year** or are used in **daily business operations**.

Key Characteristics:

- **Short-term** (usable or convertible within 12 months)
- **Indicate liquidity** (how quickly the company can access cash)
- **Meant for sale or consumption**

Examples:

- Cash in hand
- Accounts receivable (money owed by customers)
- Inventory (stock of goods)
- Bank balance

Example Sentence:

"Cash available in the company's drawer is a current asset."

Understanding Financial Support Options: Grants, Loans, and Equity Capital

1. Grant

Definition:

A grant is financial assistance provided by organizations, governments, or donor agencies that **does not need to be repaid**. It is typically awarded for research, education, startups, or public welfare projects.

Key Features:

- **Non-repayable** (no refund required)
- **Purpose-specific** (must be used as intended)
- May come with **conditions** set by the donor
- Available to **businesses and individuals**

Example:

A government provides an IT startup with **Rs. 500,000** as a grant to develop a free education platform.

2. Loan

Definition:

A loan is borrowed money from a bank or financial institution that must be **repaid with interest** within a fixed period.

Key Features:

- **Repayment required** (with interest)
- Fixed **installment payments**
- May require **collateral** (property, assets)
- Used for business expansion, personal needs, etc.

Example:

A software company takes a **Rs. 2 million** loan to expand operations, repayable in **3 years** with interest.

3. Equity Capital

Definition:

Equity capital is funding provided by **investors or shareholders** in exchange for **ownership shares** in the business. Investors earn profits through dividends and gain decision-making rights.

Key Features:

- **No repayment obligation**
- Investor receives **ownership stake**
- Share in **profits (dividends)**
- Investor **bears financial risk**

Example:

An investor contributes **Rs. 1 crore** to a software company in exchange for **20% ownership shares**.

The Three Fundamental Financial Statements

1. Income Statement (Profit & Loss Statement)

Definition:

The Income Statement shows a company's **revenues and expenses** over a specific period (month, quarter, or year), determining whether the business made a **profit or loss**.

Purpose:

- Measures **profitability** (Is the business making money?)
- Evaluates **operational performance**
- Helps investors assess **business growth**

Key Components:

- **Revenue (Sales)** – Total income from goods/services
- **Cost of Goods Sold (COGS)** – Direct production costs
- **Gross Profit** = Revenue – COGS
- **Operating Expenses** – Salaries, rent, marketing
- **Net Profit/Loss** = Gross Profit – Operating Expenses

Example:

If a company earns **Rs. 1 crore** in revenue and spends **Rs. 70 lakh** on expenses, its **net profit is Rs. 30 lakh**.

2. Balance Sheet

Definition:

The Balance Sheet provides a **snapshot** of a company's financial position on a specific date, listing its **assets, liabilities, and owner's equity**.

Purpose:

- Shows **what the company owns vs. owes**

- Assesses **financial stability & solvency**
- Helps investors analyze **debt-to-asset ratio**

Key Components:

- **Assets** (What the company owns)
  - *Current Assets*: Cash, inventory, receivables
  - *Fixed Assets*: Buildings, machinery
- **Liabilities** (What the company owes)
  - *Short-term*: Payables, short-term loans
  - *Long-term*: Bank loans, mortgages
- **Owner's Equity** = Assets – Liabilities

Basic Formula:

Assets = Liabilities + Owner's Equity

Example:

If a company has **Rs. 50 lakh** in assets and **Rs. 20 lakh** in liabilities, its **equity is Rs. 30 lakh**.

3. Cash Flow Statement

Definition:

This statement tracks **cash inflows and outflows**, showing how cash moves through **operations, investments, and financing**.

Purpose:

- Reveals **actual cash availability** (Profit ≠ Cash)
- Helps plan **payments, salaries, and bills**
- Measures **liquidity** (Can the business cover expenses?)

Key Components:

- **Operating Activities** (Cash from sales, salaries, suppliers)
- **Investing Activities** (Cash spent on equipment, property)
- **Financing Activities** (Loans, dividends, stock sales)

Example:

If a company earns **Rs. 10 lakh** from sales, spends **Rs. 4 lakh** on equipment, and repays **Rs. 2 lakh** in loans, its **net cash flow is Rs. 4 lakh**.

Role of Each Statement in Financial Health Assessment

Statement	What It Reveals
Income Statement	Is the business profitable? Shows revenue growth and operational efficiency.
Balance Sheet	Is the company financially stable? Compares assets vs. debts.
Cash Flow Statement	Does the business have enough cash to operate? Tracks real liquidity.

What is a Computer/Software Contract?

A computer contract (or software contract) is a **legally binding agreement** between a software developer/vendor and a client/customer. It clearly defines:

- Rights & responsibilities of both parties
- Project deliverables & deadlines
- Payment terms
- Legal protections

These contracts are used when:

1. A company hires a software firm to develop **custom software**

2. Purchasing, licensing, or maintaining software

Client Obligations in a Software Contract

What Must the Client Do?

1. **Provide Clear Requirements**
   - Must specify **functional needs** for accurate development.
2. **Give Timely Feedback**
   - Review delivered versions/modules promptly for improvements.
3. **Supply Necessary Resources**
   - Provide required **hardware, databases, or third-party tools**.
4. **Make Payments on Time**
   - Follow the contract's **payment schedule** (advance, milestones, final).
5. **Maintain Confidentiality**
   - Cannot share the developer's **proprietary tools/designs** with others.
6. **Participate in Testing**
   - Must be involved in **acceptance testing** before final approval.

Key Elements of a Custom Software Contract

Must-Have Clauses in Every Agreement

Clause	Purpose
1. Scope of Work	Defines software features, functional & non-functional requirements.
2. Timelines & Milestones	Sets deadlines for design, coding, testing, and deployment.
3. Payment Terms	Specifies total cost, advance payments, and milestone-based installments.
4. Confidentiality Clause	Ensures neither party shares sensitive business/technical data.
5. Intellectual Property (IP) Rights	Determines who owns the final software (full ownership or licensing).
6. Testing & Acceptance Criteria	Explains how the client will test and approve the software.
7. Maintenance & Support	Covers post-delivery bug fixes, updates, and support duration.
8. Termination Clause	Outlines penalties if either party cancels the contract prematurely.
9. Dispute Resolution	Defines how conflicts will be resolved (arbitration, mediation, or court).

Intellectual Property Rights (IPR) in Software Development

What Are Intellectual Property Rights (IPR)?

Intellectual Property Rights (IPR) are **legal protections** granted to individuals or companies for their original creations. These rights apply to various fields, including:

- Software
- Music
- Books
- Designs
- Inventions
- Brand names & logos

Just as land has an owner, **ideas and creations** also have owners—IPR legally protects this ownership.

Types of Intellectual Property Rights in Software

1. **Copyright** – Protects **source code and design**
2. **Patent** – Covers **innovative algorithms or processes**
3. **Trademark** – Protects **software names, logos, and branding**
4. **Trade Secret** – Keeps **confidential formulas/methods** private

Subcribe my youtube channel Habiba's Lectures for video lectures

What Can and Cannot Be Patented?

What Can Be Patented?

- **New** (world's first)
- **Inventive** (not obvious)
- **Useful** (practical application)

Examples in Software:

Unique algorithms

Innovative data processing methods

Advanced authentication techniques

Special hardware-software integration

What Cannot Be Patented?

Mathematical formulas

Abstract ideas

Natural laws

Obvious features (e.g., basic login system)

Graphical layouts (protected by copyright instead)

How to Protect Software IP Rights

1. Copyright Registration

- The most common method
- Grants legal ownership of the code
- Allows legal action against unauthorized copying

2. Software Patents (Selective Cases)

- Protects **unique software processes**
- Example: Google's **PageRank algorithm** was patented

3. Trade Secrets

- Keeps **confidential algorithms** hidden
- Example: Coca-Cola's secret formula

4. License Agreements (EULA)

- Legally binds users to prevent misuse
- Example: **End User License Agreements**

5. Trademark Protection

- Protects **software names & logos**

- Example: **Windows™, Oracle®**

6. Digital Rights Management (DRM) & Watermarking

- Prevents unauthorized distribution

Permitted Acts Under Copyright Law

While copyright restricts unauthorized use, some exceptions exist:

Permitted Act	Description	Example
Fair Use	Limited copying for education/research	Quoting code in a research paper
Backup Copy	One backup for personal use	Keeping a Windows ISO backup
Decompilation for Interoperability	Reverse-engineering for compatibility	Analyzing APIs for database integration
Temporary RAM Copy	Automatic copies during execution	Software running in memory
Open Source License	Allows modification under license terms	Using & modifying Linux code

Human Resource Management (HRM) in IT & Changing Management Practices

What is HRM?

Human Resource Management (HRM) refers to the **strategic practices** used to recruit, train, manage, and motivate employees within an organization. In a software company, employees (developers, testers, designers) are the most valuable assets—HRM ensures they are managed effectively.

Key HRM Concepts in IT

1. Performance Appraisal System

Definition:

A structured process where an organization evaluates employee performance to measure productivity, identify training needs, and determine promotions/salary increments.

Purpose:

Track employee performance
Decide promotions & salary hikes
Identify training requirements

Boost motivation through feedback

Spot underperformance

Example:

A developer who completes **3 major projects with zero bugs** in a year may receive a **high appraisal score**, leading to a **promotion or bonus**.

2. Equal Employment Opportunity (EEO)

Definition:

EEO ensures **fair job opportunities** for all individuals—regardless of gender, race, religion, or disability.

How to Ensure EEO?

Implement **non-discriminatory hiring/promotion policies**

Maintain a **transparent recruitment process**

Build a **diverse workforce**

Train HR staff on **inclusive practices**

Example:

A job posting stating:

"We are an Equal Opportunity Employer—women, differently-abled individuals, and minorities are encouraged to apply."

Subcribe my youtube channel Habiba's Lectures for video lectures

Functions of HRM in a Software Company

Function	Description	Example
Recruitment & Selection	Hiring skilled developers/designers	Hiring a React.js developer via LinkedIn
Training & Development	Upskill employees on new technologies	AI/Blockchain workshops
Performance Appraisal	Evaluating employee contributions	Annual reviews with ratings
Compensation & Benefits	Managing salaries, bonuses, perks	Offering stock options & remote work
Workplace Policies	Setting rules for work culture	Hybrid work policies post-COVID
Employee Engagement	Keeping staff motivated	Hackathons, team-building events
Handling Resignations/Layoffs	Managing exits professionally	Conducting exit interviews
IT in HRM	Using HR software for efficiency	Tools like Zoho People, BambooHR

Discrimination in HRM: Forms & Impact

What is Discrimination?

Unfair treatment of employees/applicants based on **personal traits** rather than skills.

Common Forms:

1. **Gender Discrimination** – Denying promotions to women
2. **Racial/Ethnic Bias** – Rejecting candidates based on background
3. **Age Discrimination** – Not hiring older professionals
4. **Religious Bias** – Denying prayer breaks

5. **Disability Discrimination** – Ignoring qualified differently-abled candidates

Negative Impacts:

Low morale (employees feel undervalued)
High turnover (talented staff leave)
Legal risks (lawsuits damage reputation)
Loss of talent (skilled candidates rejected unfairly)
Toxic work culture (reduced teamwork & trust)

Health and Safety in the Workplace and System Safety

1. Health and Safety at Work

What Does It Mean?

Health and safety in the workplace ensures that employees work in an environment where:

- They are **physically safe**

- They experience **minimal stress**
- Their workspace is **comfortable** (ergonomic chairs, proper lighting, etc.)

Example:

If a developer sits on a **low-quality chair for 8 hours**, they may develop back pain. That's why companies invest in **ergonomic furniture**.

2. System Safety

What Does It Mean?

System safety refers to designing **secure and reliable** software and hardware systems that protect user and company data from risks.

Example:

If **banking software** is not properly secured, customer funds could be stolen. That's why every system must include **testing, validation, and error-handling**.

Long Question: Factors Affecting System Safety & Ensuring Safety in SDLC

Factors Affecting System Safety

1. **Human Error**
   - Mistakes in coding or design (e.g., missing input validation, leading to hacking risks).
2. **Weak Design**
   - Poorly structured software that fails under load or attacks (e.g., no two-factor authentication in login systems).
3. **Lack of Testing**
   - Bugs remain undetected if software isn't tested properly (e.g., system crashes when users enter unexpected characters).
4. **Outdated Technology**
   - Old frameworks lack security patches and compatibility with new features.
5. **Security Loopholes**
   - Missing access controls or encryption makes systems vulnerable to hacking.
6. **Lack of Team Training**
   - Developers unaware of new threats make avoidable mistakes.
7. **Poor Documentation**
   - New developers struggle to understand the system, leading to errors.

Ensuring Safety in SDLC (Software Development Life Cycle)

Each phase of SDLC must incorporate safety measures:

SDLC Phase	Safety Measures	Example
Requirements	Define security needs	"System should lock after 3 wrong password attempts."
Design	Secure architecture & threat modeling	Multi-layer login (password + OTP + account lock).
Development	Secure coding practices	Using parameterized queries to prevent SQL injection.
Testing	Manual & automated security tests	Tools: Selenium, JUnit, OWASP ZAP.
Deployment	Secure server setup	Firewalls, SSL certificates.
Maintenance	Regular updates & logs	Patching vulnerabilities, monitoring logs.
Documentation	Clear system guides	Helps future developers understand security measures.
Training	Regular security awareness	Keeping teams updated on threats.

Software Liability and Computer Misuse

1. Software Liability

Software liability refers to the legal responsibility of developers or companies when their software causes harm, such as:

- **Data loss**
- **Financial damage**
- **System failures**

Example:

If a bug in **hospital management software** leads to incorrect medicine prescriptions, the developer could face legal consequences.

2. Liability and Practice in IT

This determines **who is legally responsible** (criminally or civilly) when an IT product causes harm to individuals or businesses.

3. Criminal Law in IT

Covers illegal activities involving computers, such as:

- **Hacking**
- **Virus distribution**
- **Data theft**

These are punishable under cybercrime laws.

Computer Misuse: Definition and Categories

Definition:

Computer misuse refers to **unauthorized, illegal, or unethical use** of computers or networks, causing harm or security breaches.
(Example: UK's Computer Misuse Act 1990)

Categories of Computer Misuse with Examples:

Category	Description	Example
Unauthorized Access	Illegally entering a system without permission	A student hacking a university portal to alter grades
Access with Criminal Intent	Breaking into systems to commit fraud, blackmail, or leaks	A hacker stealing bank details to transfer money
Unauthorized Data Modification	Altering software/data without permission	Defacing a website or injecting a virus
Malware Creation/Distribution	Developing/spreading harmful software	Creating ransomware that locks systems
Denial of Service (DoS) Attacks	Overloading servers to crash websites	Taking down an e-commerce site during a sale
Data/Identity Theft	Stealing personal or financial information	Hacking Facebook accounts to send fake messages
Cyberstalking/Harassment	Repeated online threats or abuse	Sending threatening emails to harass someone

Legal Consequences (Worldwide & Pakistan)

Offense	Possible Punishment
Unauthorized Access	1-3 years jail or heavy fine

Offense	Possible Punishment
Hacking with Criminal Intent	5-10 years jail
Data Tampering	3-7 years jail
Malware Creation	Up to 14 years jail
Identity Theft	3 years jail + fine
Cyber Harassment (PECA Act)	1-3 years jail + Rs. 1 million fine

In Pakistan:

- **PECA 2016 (Prevention of Electronic Crimes Act)** covers cybercrimes.
- **FIA (Federal Investigation Agency)** handles cybercrime complaints.

Informed Consent and Privacy Protection

What is Informed Consent?

Informed consent refers to obtaining explicit permission from individuals before collecting their personal data, while clearly explaining:

- **What data** is being collected

- **Why** it's needed
- **How** it will be used
- **Who** it will be shared with

This ensures users **voluntarily agree** after understanding all implications.

Key Principles of Informed Consent

Principle	Description
Transparency	Clearly explain what data is collected and why
Purpose Limitation	Specify exact usage (research, services, etc.)
Storage Duration	State how long data will be retained
Right to Withdraw	Users can revoke consent anytime
Voluntary Agreement	No coercion—users must freely agree

How Informed Consent Protects Privacy

1. **User Control**
   - Individuals decide what data they share.
2. **Prevents Misuse**
   - Organizations **cannot** use data beyond agreed purposes.
3. **Legal Accountability**
   - If violated, users can take legal action (e.g., under **GDPR**).
4. **Compliance Requirement**

- Many laws (e.g., **EU's GDPR, Pakistan's PECA**) mandate consent.

Real-World Example

When a **mobile app** requests location access:

Informed Consent Example:

- *"This app collects your location to provide real-time weather updates. You can Allow or Deny access."*

Privacy Violation:

- If the app **secretly** tracks location without explanation.

Why It Matters?

- Empowers users to **protect their data**
- Holds companies **accountable** for ethical data practices
- Required by **privacy laws worldwide**

By ensuring proper informed consent, we balance **data utility** with **individual privacy rights**.

Professional Ethics and Codes of Conduct in Computing

Introduction to Computing Ethics

Computing professionals - including software engineers, system developers, and data scientists - bear not just technical responsibilities but also significant ethical obligations. Their work directly impacts society, making adherence to ethical codes a professional requirement.

Major Professional Codes of Conduct

1. BCS (British Computer Society) Code of Conduct

The BCS is a UK-based professional body that establishes standards for computer science and IT professionals.

Key Principles:

- Prioritize public interest (users, society, environment)
- Maintain integrity (honesty, transparency)
- Continuously develop professional competence
- Support colleagues' professional growth

2. IEEE Code of Ethics

The Institute of Electrical and Electronics Engineers sets global standards for technology professionals.

Key Principles:

- Present information honestly without exaggeration
- Prioritize public safety and welfare
- Practice non-discrimination
- Promote environmental sustainability

3. ACM Code of Ethics

The Association for Computing Machinery provides ethical guidelines for computing professionals worldwide.

Key Principles:

- Respect and protect user privacy
- Avoid creating harmful systems (malware, vulnerable code)
- Honor intellectual property rights
- Participate in professional peer reviews

4. ACM-IEEE Joint Software Engineering Code

A specialized code developed collaboratively for software engineers.

Key Principles:

1. **Public:** Place public welfare first

2. **Client/Employer:** Maintain professional obligations
3. **Product:** Ensure software quality
4. **Judgment:** Exercise honest technical judgment
5. **Management:** Demonstrate ethical leadership
6. **Profession:** Advance the field's reputation
7. **Colleagues:** Treat peers fairly
8. **Self:** Commit to continuous improvement

Protecting Public Interest: Professional Responsibilities

Computing professionals safeguard public interest through these key practices:

1. **Developing Safe and Reliable Systems**

- Create secure, dependable software
- Example: Hospital systems must be fail-safe to protect lives

2. **Ensuring Data Privacy**

- Implement strong encryption
- Obtain informed consent for data collection
- Prevent unauthorized data sharing

3. **Addressing Security Vulnerabilities**

- Promptly fix discovered bugs

- Report critical vulnerabilities to proper authorities

4. **Preventing Harmful Applications**

- Avoid designing systems for misinformation or harassment
- Ensure socially responsible technology

5. **Considering Environmental Impact**

- Develop energy-efficient software
- Apply green computing principles

6. **Maintaining Transparent Communication**

- Provide accurate, non-misleading information
- Example: Health apps must report correct data