Software Engineering – 10 Mark Answers

1. (A)Define Software Engineering.

Answer:

Software Engineering is a disciplined approach to designing, developing, testing, and maintaining software using engineering principles. It ensures that software is reliable, efficient, cost-effective, and meets user requirements.

(B) Explain how software engineering differs from traditional engineering.

Answer:

- Traditional engineering deals with physical systems; software engineering deals with logical, intangible systems.
- Software requirements change frequently; traditional requirements are stable.
- Software doesn’t wear physically but becomes outdated without maintenance.
- Development is more iterative and user-feedback based.

(C) Discuss the importance of software product, software process, and key software attributes in building high-quality systems.

Answer:

- **Product:** Defines what the software delivers.
- **Process:** Defines how the software is developed.
- **Key attributes:** Reliability, maintainability, usability, security, efficiency.

  Together, they ensure high-quality software that satisfies users.

2. (A)What are software products and software processes?

Answer:

- **Software products:** Deliverables like apps, OS, websites.
- **Software processes:** Activities like requirements, design, coding, testing, maintenance.

(B) Compare generic and customized software products.

Answer:

- **Generic:** Made for large market (MS Office).
- **Customized:** Developed for specific clients (hospital ERP).

  Generic = cheaper; customized = tailored.

(C) Describe the main process activities in a software process with suitable examples.

Answer:

1. **Specification** → collecting requirements.
2. **Design/Implementation** → architecture + coding.
3. **Validation** → testing.
4. **Maintenance** → updates, bug fixes.

---

3. (A) Describe the key software attributes: reliability, maintainability, usability, efficiency, security, and acceptability.

Answer:

- Reliability → no failures
- Maintainability → easy to update
- Usability → user-friendly
- Efficiency → fast & optimized
- Security → protects data

- Acceptability → fits user needs

(B) Explain with examples how each attribute influences software quality and user satisfaction.

Answer:

Reliable apps build trust, secure apps protect data, efficient apps perform better, usable apps reduce user frustration, and maintainable apps are easier to improve.

4. (A) Explain the major Software Engineering challenges in modern software development.

Answer:

High complexity, fast tech changes, security needs, scalability demands, tight deadlines, multi-team coordination, and integrating different platforms.

(B) Discuss challenges related to globalization, legacy system integration, team coordination, and handling rapidly changing user requirements.

Answer:

- Global teams → communication issues
- Legacy systems → outdated tech
- Coordination → requires tools & planning
- Changing requirements → needs Agile adaptability

5. (A) Describe the Software Development Life Cycle (SDLC).

Answer:

SDLC is the structured approach to building software from requirements to maintenance.

(B) Explain each phase in detail and illustrate how proper requirement analysis, design, implementation, testing, deployment, and maintenance ensure overall system success.

Answer:

- Requirement analysis
- Design
- Implementation
- Testing
- Deployment
- Maintenance

Each stage ensures reliability, correctness, and long-term success.

---

6. (A)Explain the Waterfall Model, Spiral Model (Boehm’s), and Iterative Enhancement Model.

Answer:

- **Waterfall:** Sequential model.
- **Spiral:** Cycles with risk analysis.
- **Iterative:** Develop in small increments.

(B)Compare them in terms of risk handling, requirement stability, customer involvement, and project size suitability.

Answer:

- Risk: Spiral > Iterative > Waterfall
- Requirements: Waterfall needs fixed; Spiral/Iterative allow changes
- Customer involvement: Spiral high
- Project size: Waterfall small; Spiral large; Iterative all

---

7. (A) Provide a detailed overview of Risk Management in software engineering.

Answer:

Risk management identifies, analyzes, plans, and monitors risks to avoid project failure.

(B)Explain risk identification, classification, risk assessment, risk planning, risk monitoring, and mitigation strategies with examples.

Answer:

- Identification → listing risks
- Classification → technical, financial
- Assessment → probability & impact
- Planning → backup strategies
- Monitoring → track risks
- Mitigation → actions like prototypes, extra training

---

8. (A) Define Project Management in software engineering.

Answer:

Project management ensures software is developed within time, budget, and quality limits.

(B)Discuss project planning, scheduling, cost estimation, resource allocation, team management, and project monitoring.

Answer:

- Planning → goals
- Scheduling → timeline
- Cost estimation → budget
- Resources → assigning tools & people
- Team management → coordination
- Monitoring → tracking progress

(C)Explain how process visibility helps in managing large-scale projects.

Answer:

It shows progress clearly, prevents delays, and improves decision-making.

---

9. (A)Discuss professional and ethical responsibilities of software engineers.

Answer:

Engineers must follow honesty, confidentiality, competence, integrity, respect for IP, and professionalism.

(B) Explain issues related to confidentiality, intellectual property rights, competence, conflicts of interest, and the importance of professional codes of ethics.

Answer:

- Confidentiality → protect client data
- IPR → respect patents/copyright
- Competence → work only in skilled areas
- Conflicts → avoid personal bias
- Ethics codes → follow ACM/IEEE rules

---

10. (A) What is System Engineering?

Answer:

System engineering deals with designing and managing large systems involving hardware, software, people, and processes.

(B) Explain system concepts such as system environment, system modeling, system procurement, and the system engineering process.

Answer:

- Environment → surrounding systems
- Modeling → diagrams and representations
- Procurement → acquiring system components
- Process → requirement to deployment

(C) Discuss system architecture modeling, human-factor considerations, and the role of system reliability engineering.

Answer:

- Architecture modeling → structure of components
- Human factors → usability & ergonomics
- Reliability engineering → ensures fewer failures