No. of Printed Pages : 4 **MCS-213**

MASTER OF COMPUTER APPLICATIONS (MCA) (NEW)

Term-End Examination

June, 2023

MCS-213 : SOFTWARE ENGINEERING

Time : 3 Hours *Maximum Marks : 100*

Weightage : 70%

Note* :** (*i*) *Question No.* ***1 *is compulsory.*

(*ii*) *Attempt any* ***three*** *questions from the rest.*

1. (a) Explain the following approaches for the development of mobile applications : 10

(i) Native Application Development

(ii) Rapid Mobile Application Development (RMAD)

(iii) Progressive Web Applications (PWAs)

Also, mention any *two* advantages for each.

(b) Discuss the (i) requirements related optimization and (ii) architecture and design related optimizations in context of First Time Right (FTR) framework. 10

(c) In context to software project estimation, explain the following (highlighting their main tasks) : 10

(i) Estimating the project-size

(ii) Estimating efforts

(iii) Estimating the schedule

(iv) Estimating the total cost

(d) Discuss the following software engineering models : 10

(i) Waterfall model

(ii) Spiral model

2. (a) Discuss the Human Computer Interface (HCI) and User Experience (UX) and designing for mobility aspects of software design phase. 10

(b) Explain defect metrics and maintainability metrics for measurement of software quality. 10

3. (a) Define cleanroom software engineering. List and explain the principles for the cleanroom based software development. 10

(b) Explain the following emerging trends in software engineering highlighting their salient features, tools, technologies, purpose of usage and advantages : 10

(i) Low Code and No Code platforms

(ii) Containerization

4. (a) List and discuss the issues and challenges in management of web-based projects. 10

(b) Define CASE tools. What are the various categories of CASE tools available ? Also mention the factors that affect their deployment in an organisation. 10

5. Write short notes on the following : 4×5=20

(a) Cloud platforms

(b) Continuous Delivery Model

(c) COCOMO Model

(d) Control Flow Graph (CFG) along with an example graph for any programming construct.