National Institute of Technology, Rourkela

Name of the Examination: M. Tech./PhD Mid-Semester (Autumn 2021-22)

Branch	**: CS**	**Semester**	**: I**
Title of the Course	**: Advanced Software Engineering**		
Course Code	**: CS6401**		
Time	**: 2 Hours**	**Maximum Marks: 30**	

General Instruction :

1. Attempt all questions in **Section A** and **Section B.**

Section-A **(20 x 0.5 = 10)**

(Attempt all of the MCQ questions)

1. This section contains 20 Multiple Choice Questions. Each question carries 0.5 Marks.

Section-B **(6 +4+ 5 +5= 20)**

(Attempt all of the following questions)

1. Describe the process of specifying a major piece of software: the main documents produced, what are the contents we should contain in those documents, their immediate purpose and their on-going role in the software life cycle. **(6 Marks)**

2. Describe the role that formal methods can play at each stage of the software life cycle. Explain any disadvantages of the uses of formal methods that you have discussed. **(4 Marks)**

3. **(2+1+2=5 Marks)**

(i) Outline the similarities and differences of the V-model with the iterative waterfall model?

(ii) Give an example of a development project for which V-model can be considered appropriate and also give an example of a project for which it would be clearly inappropriate.

(iii) Identify how exactly risk is handled in both the prototyping model and spiral model. How do these two models compare with each other with respect to their risk handling capabilities?

4. **(2+3 =5 Marks)**

(i) What write the pre- and post-conditions to axiomatically specify a function that will find the difference between the largest and the smallest values held in an integer array.

(ii) Algebraically specify an abstract data type that stores a set of elements and supports the following operations. Assume that the ADT element has already been specified and you can use it:

New: creates a null set.
Add: takes a set and an element and returns the set with the additional elements stored.
Size: takes a set as argument and returns the number of elements in the set.
Remove: takes a set and an element as its argument and returns the set with the element removed.
Contains: takes a set and an element as its argument and returns the Boolean value true if the element does belongs to the set and returns false if the element does not belong to the set.
Equals: takes two sets as arguments and returns true if they contain identical elements and returns false otherwise.