An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF AIML

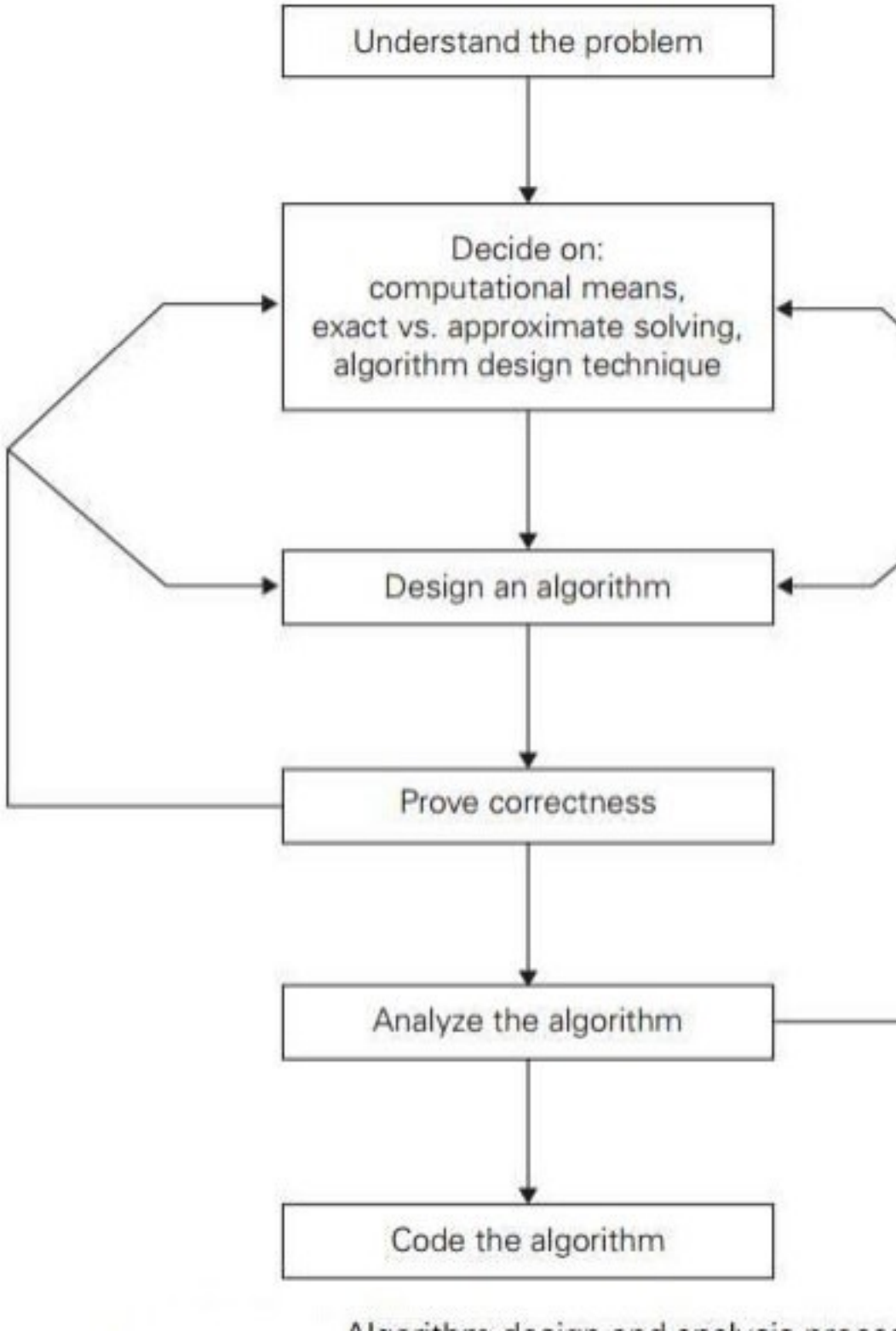
OBLEM SOLVING AND C PROGRAMMING

I YEAR - I SEM

T 1 – Introduction to Problem Solving Techniques

TOPIC 5 – Algorithmic Problem Solving

ic problem solving is solving
em that require the formulation
rithm for the solution.



rocess of **finding the input** of the problem that the algorith

important to specify **exactly the set of inputs** the alg

algorithm is not one that works most of the time, but
for all legitimate inputs.

ing the Capabilities of the Computational Device:
structions are executed one after another, it is called **seque**
structions are executed concurrently, it is called **parallel a**

PROBLEM SOLVING

rincipal decision is to choose between solving the pro
pproximately.
his, the algorithms are classified as **exact** algorithm and

data structure:
structure plays a vital role in designing and analysis the alg
e of the algorithm design techniques also depend on the
fying a problem's instance.

Algorithm+ Data structure=programs.

hm design technique (or “strategy” or “paradigm”) is a gen
problems algorithmically that is applicable to a variety
ent areas of computing.
hese techniques is of utmost importance for the following r

provide guidance for designing algorithms for new problem

thms are the cornerstone of computer science

e:

docode is a mixture of a natural language and programming langu
tructs.

docode is usually more precise than natural language, and its usag
e success in algorithm descriptions.

e earlier days of computing, the dominant vehicle for specifying a
chart, a method of expressing an algorithm by a collection of conn
es containing descriptions of the algorithm's steps.

ing language:

gramming language can be fed into an electronic computer directly
ead, it needs to be converted into a computer program written in a
puter language.

can look at such a program as yet another way of specifying the al
preferable to consider it as the algorithm's implementation.

orithm has been specified, you have to prove its correctness.

have to prove that the algorithm **yields a required result for ever**
nite amount of time.

echnique for proving correctness is to use **mathematical inductio**
iterations provide a natural sequence of steps needed for such proc

worth mentioning that although tracing the algorithm's performanc
uts can be a very worthwhile activity, it cannot prove the algorithm
.

to show that an algorithm is incorrect, you need just one instance
gorithm fails.

gorithms are destined to be ultimately implemented as
.

ming an algorithm presents both a peril and an opportunity.

ng program provides an additional opportunity in all
analysis (Pattern and Observations) of the underlying algo

analysis is based on timing the program on several inputs
g the results obtained.

cy.

fficiency:

ating how fast the algorithm runs.

efficiency:

ating how much extra memory it uses.

cy.

orithm should be precisely defined and investigated with

natical expressions.

er algorithms are easier to understand and easier to program

e algorithms usually contain fewer bugs.