SET 1

1. What realities necessitated an engineering approach to software development? **3**

   • Number of people who have an interest in the features and functions provided by a specific application has grown dramatically. Each of them has a slightly different idea of what software features and functions. There is need to understand the problem before a software solution is developed. **1**

   • The complexity of new computer-based systems and products demands a large teams of people and also a careful attention to the interactions of all system elements. It follows that design becomes a pivotal activity. **1**

   • Strategic and tactical decision making as well as day-to-day operations and control of individuals, businesses, and governments increasingly rely on software. It follows that software should exhibit high quality. **0.5**

   • As the perceived value of a specific application grows, the likelihood is that its user base and longevity will also grow. As its user base and time-in-use increase, demands for adaptation and enhancement will also grow. It follows that software should be maintainable.

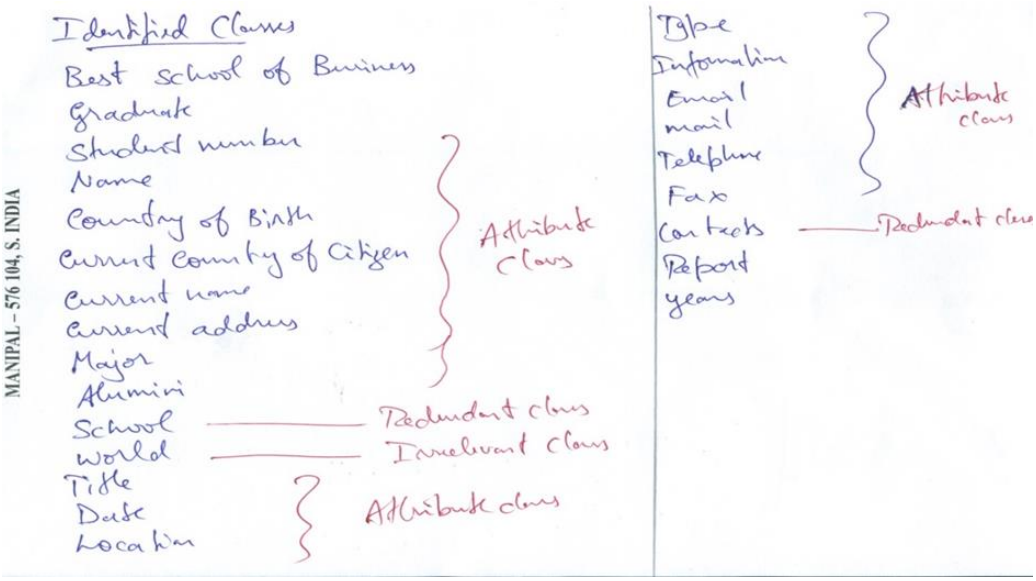
   These simple realities lead to one conclusion: software in all of its forms and across all of its application domains should be engineered. **0.5**

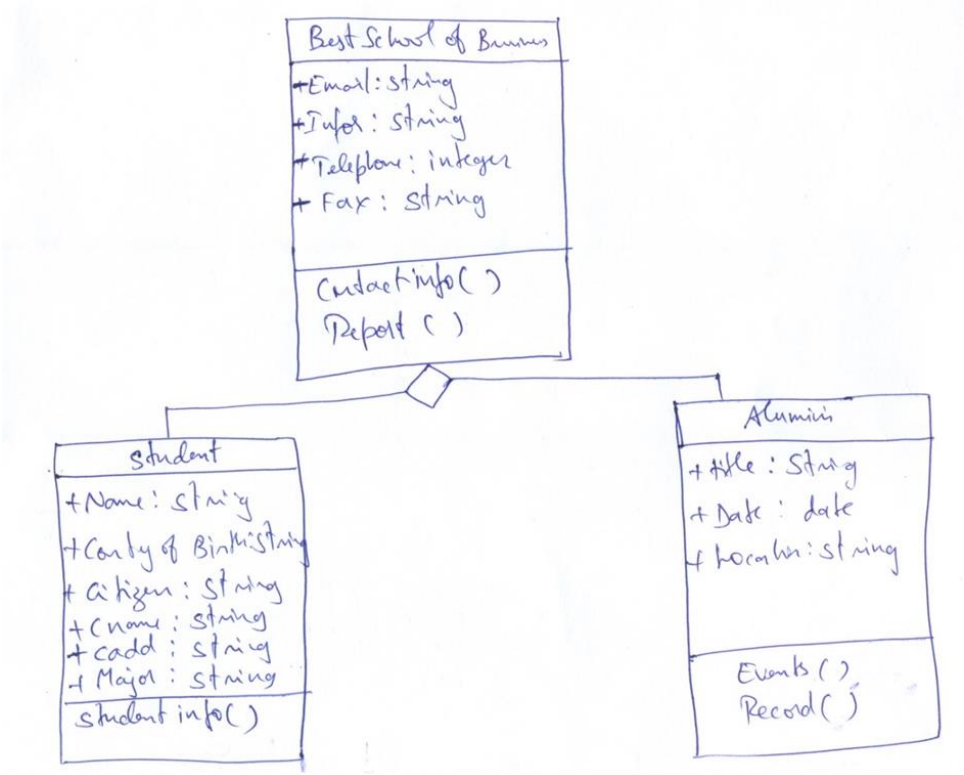
2. Identify the classes using noun phrase approach and draw the class diagrams with appropriate relationships, multiplicities etc. **4**

   The Best School of Business keeps track of each graduate's student number, name, country of birth, current country of citizenship, current name, current address, and the name of each major the student completed (each student has one or two majors). To maintain strong ties to its alumni, the school holds various events around the world. Events have title, date, location, and type (e.g., reception, dinner or seminar). The school needs to keep track of which graduates have attended which events. When a graduate attends an event, a comment is recorded about the information school officials learned from that graduate at that event. The school also keeps in contact with graduates by mail, e-mail, telephone and fax interactions. As with events, the school records information learned from the graduate from each of these contacts. When a school official knows that they will be meeting or talking to a graduate, a report is produced showing the latest information about that graduate and the information learned during the past two years from that graduate from all contacts and events the graduate attended.

   Marking Scheme: Identification of Classes using Noun Phrase Approach: 1.5 Marks

   Classes and Relations: 2.5 Marks

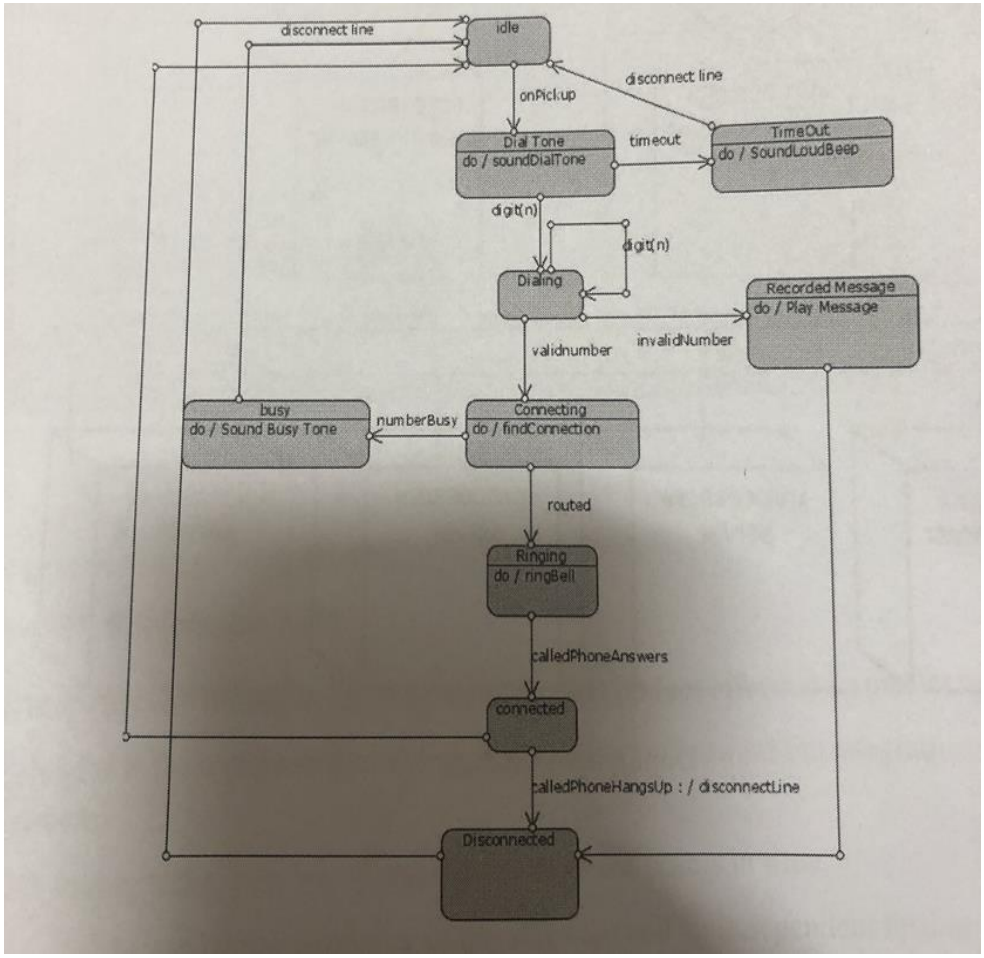




3. Draw a state diagram to model the operation of telephone line. **3**

At the beginning of the call, the telephone line is idle. When the phone receiver is picked from the hook, it gives a dial tone and can accept the dialing of digits. If after getting dial tone, if the user does not dial number within a given time interval, then time out occurs, and the phone line gets idle. After dialing a number, if the number is invalid then some recorded message is played. Upon entry of a valid number, the phone system tries to connect a call and routes it to the proper destination. If the called person answers the phone, conversation can occur. When called person hangs up, the phone disconnects and goes to idle state.

Marking Scheme: Identification of correct states and Transitions: 1.5 Marks
State diagram showing states and transitions: 1.5 Marks



4. An author publishing application is made available as a service. It is responsible for processing markdown text. It would execute a series of operations back-to-back: **3**

- Warn authors if they were using forbidden short forms such as “isn’t” or “I’m” (let’s call this the “text checker”)
- Transform Latex code to Unicode. E.g., \alpha to α (let’s call this the “latex to unicode transformer”)
- Upload the result to an S3 bucket so it is publicly available (an “S3 uploader”)

Identify the architecture pattern of this author publishing application and explain the strengths and weakness of this design.

Identification of Pipe and filter pattern (.5M)

Illustration (.5M)
Two advantages (1M)
Two Disadvantages (1M)

5. Why is it important for software processes to be agile? List the key issues stressed by an agile philosophy of software engineering. **3**

Software process provides the stability, control, and organization to an activity to prevent it from becoming chaotic. But modern software processes must only demand the activities, controls, and work products that are appropriate for the team and product to be produced – to ensure that it can accommodate changes easily and deliver a high quality software product.

Key issues: **1**

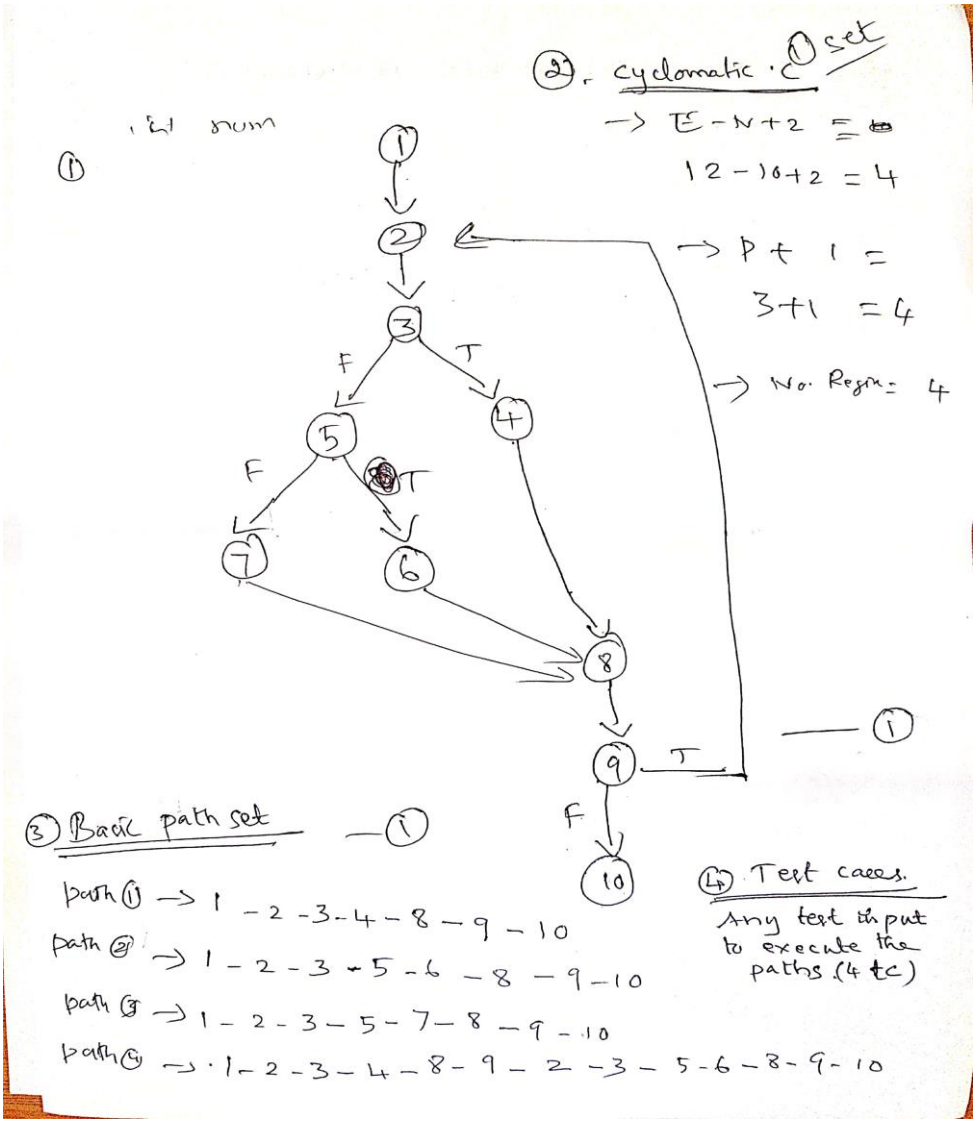
- The importance of self-organizing teams that have control over the work they perform: the team serves as its own management; Team self-organizes for the work to be done, the process to best accommodate its local environment, and the work schedule to best achieve delivery of the software increment **0.5**
- Communication and collaboration between team members and customers: adopt the customer as a part of the development team and work to eliminate the "us and them" attitude **0.5**
- Recognition that change represents opportunity: pervasiveness of change is the primary driver for agility; Support for changes should be built-in in the software engineering process as change is the heart and soul of software. **0.5**
- Emphasis on rapid delivery of software that satisfies the customer: Software increments (executable prototypes or portions of an operational system) must be delivered in short time periods so that adaptation keeps pace with change (unpredictability); de-emphasizes the importance of intermediate work products **0.5**

6. Design the test cases for the following code snippet using path testing. You are expected to follow the following steps to design the effective test case which should have the high probability of revealing the defects. **4**

1.Draw the CFG (Control Flow Graph) [1 Mark]
2.Find the Cyclomatic Complexity using three methods. [1 Mark]
3.Identify the independent paths (Basic Path Set) [1 Mark]
4.Derive test cases [1 Mark]

```
int main()
{
   int num;
   char choice;
   do {
      printf("Enter an integer number :");
      scanf("%d", &num);
      if (num == 0)
         printf("Number is ZERO.");
      else if (num > 0)
         printf("Number is POSITIVE.");
      else
         printf("Number is NEGATIVE.");
      printf("\n\nWant to check again (press Y/y for 'yes') :");
      scanf(" %c", &choice);
   } while (choice == 'Y' || choice == 'y');
```

```
        printf("\nBye Bye!!!");
        return 0;
}
```



SET 2

1. Distinguish between framework and umbrella activities. Describe the framework and umbrella activities of a generic process framework. **3**

The framework activities are applicable to all software projects, regardless of their size or complexity. On the other hand, the umbrella activities are applicable across the entire software process. Umbrella activities complement the framework activities helping a software team manage and control progress, quality, change, and risk. **0.5**

A generic process framework for software engineering encompasses five activities:

- Communication: Before any technical work can commence, it is critically important to communicate and collaborate with the stakeholders to understand stakeholders' objectives for the project and to gather requirements that help define software features and functions.
- Planning. It creates a "map" called a software project plan, that helps guide the software development team as it makes the journey. This plan defines the software engineering work by describing the technical tasks to be conducted, the risks that are likely, the resources that will be required, the work products to be produced, and a work schedule.
- Modeling. Models help to better understand both software requirements and the designs.
- Construction. This activity combines code generation and the testing that is required to uncover errors in the code.
- Deployment. The software (as a complete entity or as a partially completed increment) is delivered to the customer who evaluates the delivered product and provides feedback based on the evaluation. **1**

Typical umbrella activities include:

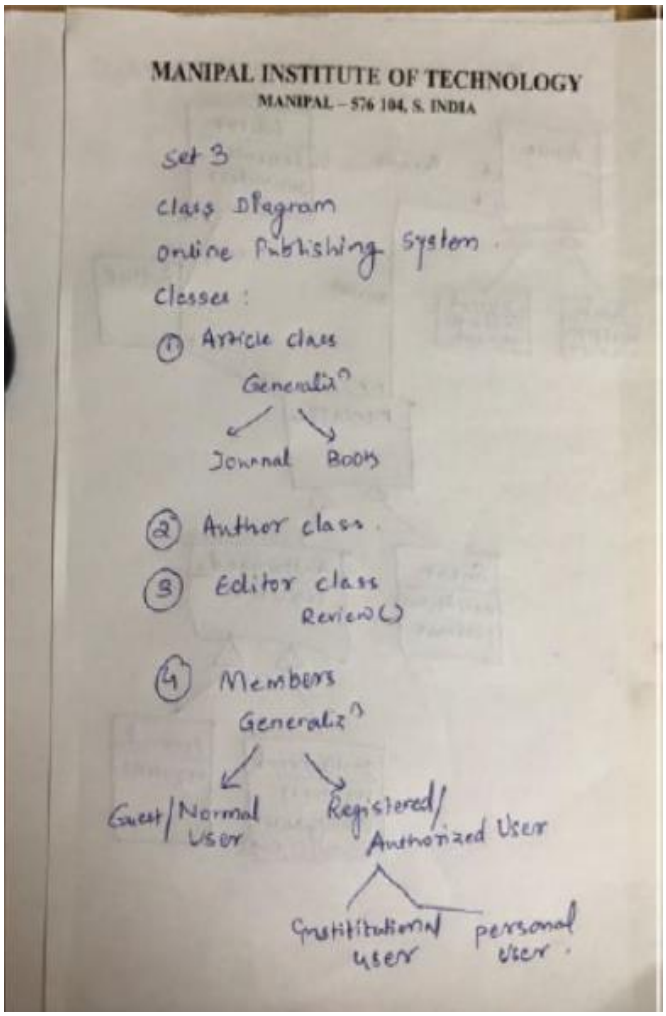
- Software project tracking and control—allows the software team to assess progress against the project plan and take any necessary action to maintain the schedule.
- Risk management—assesses risks that may affect the outcome of the project or the quality of the product.
- Software quality assurance—defines and conducts the activities required to ensure software quality.
- Technical reviews—assesses software engineering work products in an effort to uncover and remove errors before they are propagated to the next activity.
- Measurement—defines and collects process, project, and product measures that assist the team in delivering software that meets stakeholders' needs; can be used in conjunction with all other framework and umbrella activities.
- Software configuration management—manages the effects of change throughout the software process.
- Reusability management—defines criteria for work product reuse (including software components) and establishes mechanisms to achieve reusable components.
- Work product preparation and production—encompasses the activities required to create work products such as models, documents, logs, forms, and lists. **1.5**

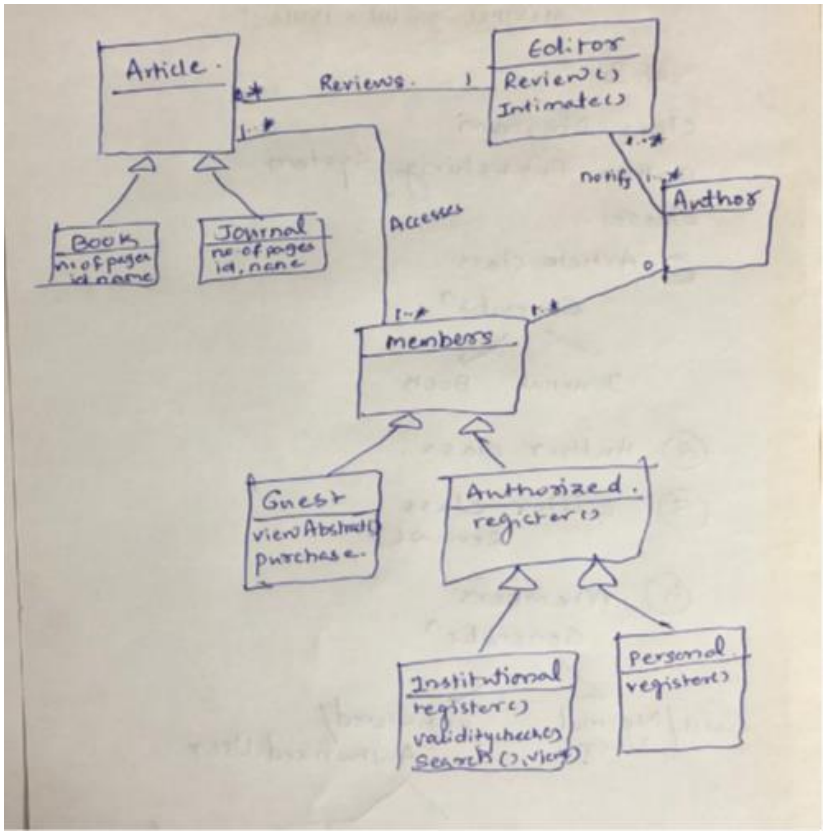
2. Identify the classes using noun phrase approach and draw the class diagrams with appropriate relationships, multiplicities etc. **4**

Online publishing system allows authors to publish their various articles. Article might be either book or journal and size should not exceed 50 pages. Authors need not be a registered user. Article will be reviewed by editor and intimation will be sent to the author. The members can search for various articles. They can view the journals published by various authors and can also buy books. Members/authorized users are of two types i.e., Institutional user and Personal user. They need to register by filling details such as Name, e-mail address and Contact information, Attestation certificates, Payment options and other details. Cost of registration for Institution is $125 for the period of 125 years. Validity of institutional user is as along as he/she is serving or studying in the institution. Institutional user can search and view for papers and books. They can buy books for $10. Personal users have period of registration of 3 years at the cost of $20. If their period of validation expires, then after which they can renew the period of registration. Password recovery option is also provided to members. Guest user can view only abstract part of the journal and can purchase the book for an amount more than the authorized user.

Marking Scheme: Identification of Classes using Noun Phrase Approach: 1.5 Marks

Classes and Relations: 2.5 Marks





3. Why are evolutionary models considered to be the best approach to software development in a modern context? What are the advantages and disadvantages of spiral and prototype models? **3**

Evolutionary models considered to be the best approach to software development in a modern context:

- Because timelines for the development of modern software are getting shorter and shorter, customers are becoming more diverse (making the understanding of requirements even harder), and changes to requirements are becoming even more common (before delivery), we need a way to provide incremental or evolutionary delivery. The evolutionary process accommodates uncertainty better than most process models, allows the delivery of partial solutions in an orderly and planned manner, and most importantly, reflects what really happens when complex systems are built. **1**

Advantages of spiral model

- Unlike other process models that end when software is delivered, the spiral model can be adapted to apply throughout the life of the computer software. Therefore, the first circuit around the spiral might represent a “concept development project” that starts at the core of the spiral and continues for multiple iterations until concept development is complete. The spiral remains operative until the software is retired. There are times when the process is dormant, but whenever a change is initiated, the process starts at the appropriate entry point (e.g., product enhancement).und the spiral might be used to represent a “product enhancement project.
- The spiral model is a realistic approach to the development of large-scale systems and software. Because software evolves as the process progresses.
- The spiral model uses prototyping as a risk reduction mechanism but, more important, enables you to apply the prototyping approach at any stage in the evolution of the product.

- It maintains the systematic stepwise approach suggested by the classic life cycle but incorporates it into an iterative framework that more realistically reflects the real world. The spiral model demands a direct consideration of technical risks at all stages of the project and, if properly applied, should reduce risks before they become problematic. The customer better understand and react to risks at each evolutionary level.

Disadvantages of spiral model:

- It may be difficult to convince customers (particularly in contract situations) that the evolutionary approach is controllable.
- It demands considerable risk assessment expertise and relies on this expertise for success. If a major risk is not uncovered and managed, problems will undoubtedly occur.

Advantages of prototype model

- The prototyping paradigm may offer the best approach when requirements are fuzzy. It serves as a mechanism for identifying software requirements Often, a customer defines a set of general objectives for software, but does not identify detailed requirements for functions and features. In other cases, the developer may be unsure of the efficiency of an algorithm, the adaptability of an operating system, or the form that human-machine interaction should take. Prototyping is the best approach in these situations. **1**
- Although prototyping can be used as a stand-alone process model, it is more commonly
- used as a technique that can be implemented within the context of any one of the process models noted in this chapter.
- A working prototype may reuse existing program fragments or tools.

Disadvantages of prototype model

- Stakeholders see what appears to be a working version of the software, unaware that the prototype is held together haphazardly, unaware that in the rush to get it working you haven't considered overall software quality or long-term maintainability. When informed that the product must be rebuilt so that high levels of quality can be maintained, stakeholders cry foul and demand that "a few fixes" be applied to make the prototype a working product. Too often, software development management relents.
- A software engineer often makes implementation compromises in order to get a prototype working quickly. An inappropriate operating system or programming language may be used simply because it is available and known; an inefficient algorithm may be implemented simply to demonstrate capability. After a time, you may become comfortable with these choices and forget all the reasons why they were inappropriate. The less-than-ideal choice has now become an integral part of the system.
- As prototype is usually a throw away system, more effort spent on it would be a waste. **1**

4. A company SDT Corp has approached you for developing a compiler. What architecture pattern would you use for this? Justify your approach. **3**

Identification of Pipe and filter pattern (.5M)

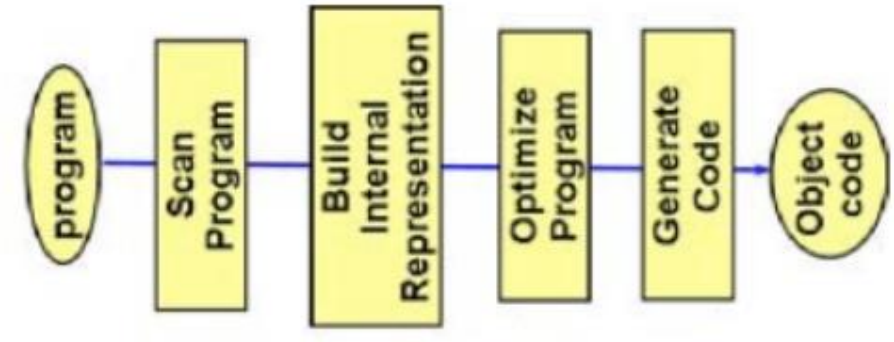
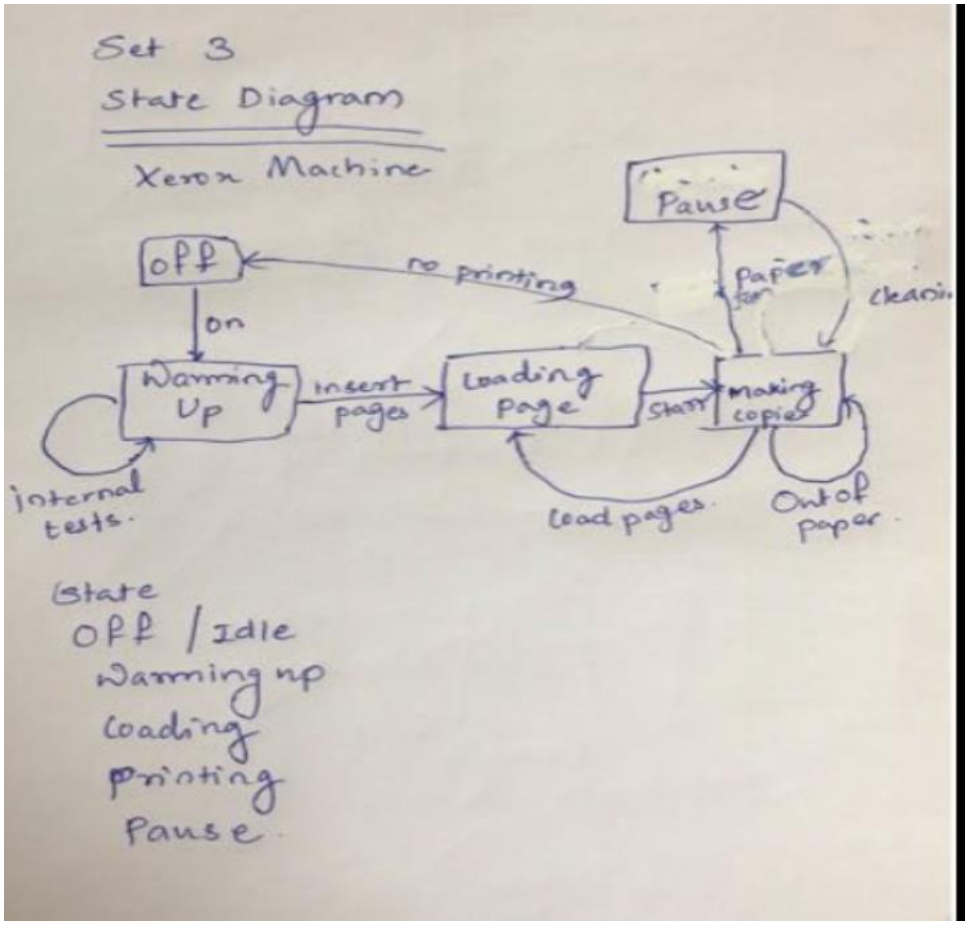


Illustration (.5M)
Two advantages (1M)
Two Disadantages(1M)

5. Identify states and events for a Photocopier (Xerox) machine from the description given below and draw the state diagram for the same. **3**

Initially the machine is off. When the operator switches on the machine, it first warms up during which it performs some internal tests. Once the tests are over, machine is ready for making copies. When operator loads a page to be photocopied and press 'start' button, machine starts making copies according to the number of copies selected. While machine is making copies, machine may go out of paper. Once operator loads sufficient pages, it can start making copies again. During the photocopy process, if paper jam occurs in the machine, operator may need to clean the path by removing the jammed paper to make the machine ready.

Marking Scheme: Identification of correct states and Transitions: 1.5 Marks
State diagram showing states and transitions: 1.5 Marks



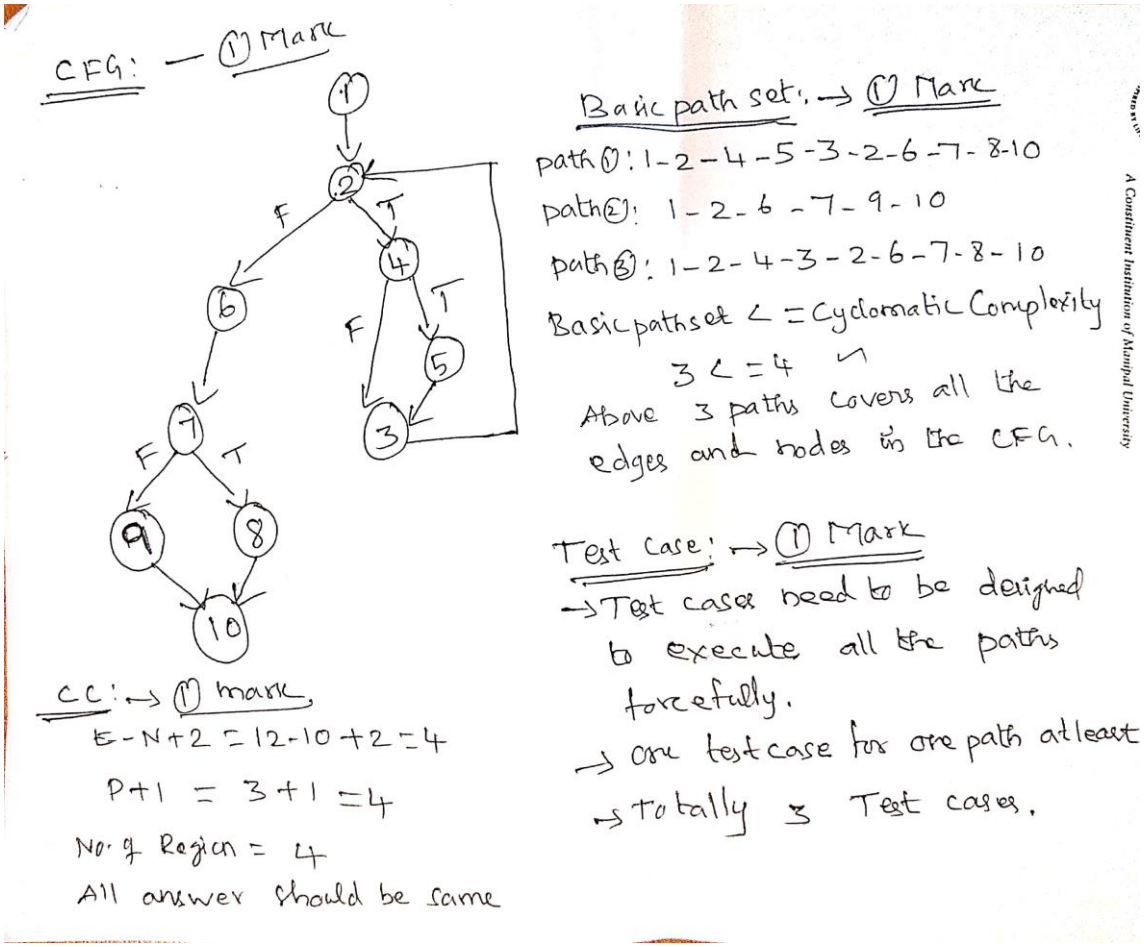
6. Design the test cases for the following code snippet using path testing. You are expected to follow the following steps to design the effective test case which have the high probability of revealing defects. **4**

1.Draw the CFG (Control Flow Graph) - 1 Mark
2.Find the Cyclomatic Complexity using three methods. - 1 Mark
3.Identify the independent paths (Basic Path Set) - 1 Mark
4.Derive test cases - 1 Mark

```
int  main()
{
 int n,i,sum;
 int mn,mx;
 printf("Input the  number : ");
 scanf("%d",&n);
   sum = 0;
printf("The positive divisor  : ");
   for (i=1;i<n;i++)
    {
    if(n%i==0)
      {
      sum=sum+i;
      printf("%d  ",i);
      }
    }
```

```
printf("\nThe sum of the divisor is : %d",sum);
   if(sum==n)
     printf("\nSo, the number is perfect.");
   else
     printf("\nSo, the number is not perfect.");
printf("\n");

return 0;
}
```



SET 3

1. What does the SCRUM emphasize to use to make the process model more effective? Discuss the development actions of the Scrum process model. **3**

Scrum emphasizes the use of a set of software process patterns that have proven effective for projects with tight timelines, changing requirements, and business criticality.

Each of these process patterns defines a set of development actions: **0.5**

- Backlog—a prioritized list of project requirements or features that provide business value for the customer. Items can be added to the backlog at any time (this is how changes are introduced). The product manager assesses the backlog and updates priorities as required.
- Sprints—consist of work units that are required to achieve a requirement defined in the backlog that must be fit into a predefined time-box14 (typically 30 days). Changes (e.g., backlog work items) are not introduced during the sprint. Hence, the sprint allows **0.5** team members to work in a short-term, but stable environment.
- Scrum meetings—are short (typically 15 minutes) meetings held daily by the Scrum team. Three key questions are asked and answered by all team members:
- What did you do since the last team meeting? **0.5**
- What obstacles are you encountering?
- What do you plan to accomplish by the next team meeting?
- A team leader, called a Scrum master, leads the meeting and assesses the responses from each person. The Scrum meeting helps the team to uncover potential problems as early as possible. Also, these daily meetings lead to "knowledge socialization" and thereby promote a self-organizing team structure.
- Demos—deliver the software increment to the customer so that functionality that has been implemented can be demonstrated and evaluated by the customer. It is important to note that the demo may not contain all planned functionality, but rather those functions that can be delivered within the time-box that was established. **0.5**

0.5

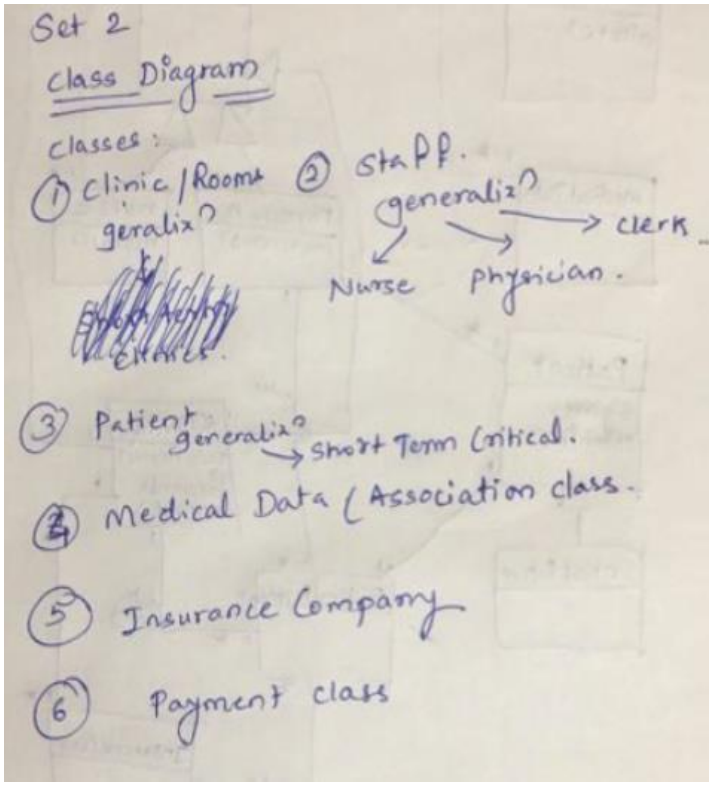
2. Identify the classes using noun phrase approach and draw the class diagrams with appropriate relationships, multiplicities etc. **4**

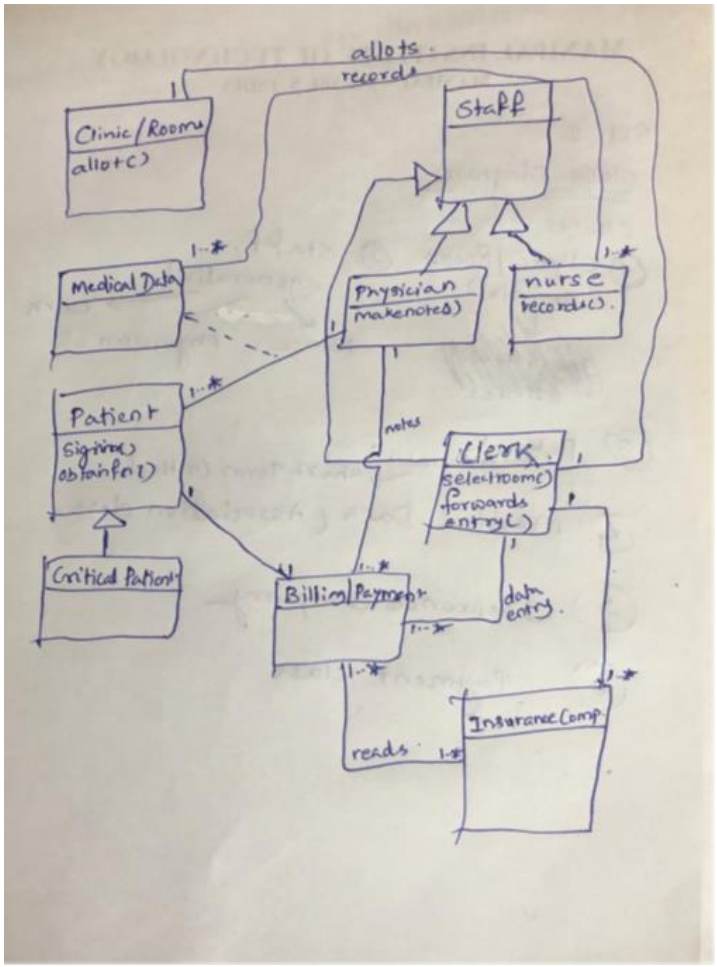
You have been hired to design a system for small health care organization. The clinic consists of several examining rooms and few rooms for short term critical care patients. A core staff of seven physicians is supplemented by internists from a local teaching hospital. Patient's medical data is stored in the central database. Patients first sign in at the front desk. Clerk checks the billing records, prints a summary status sheet and obtains file number from the system. The clerk then selects the examination room for the patient

based on the case. After waiting for the physician, the clerk moves the data packet and the patient to the examinations room. A nurse records basic medical data (weight, B.P etc.). The physician makes additional notes to both the medical and billing data and generally rights the prescription order which is given to the patient and recorded on the charts. When the patient leaves, the clerk enters the new data into the system (if any). The new billing data is forwarded to appropriate insurance company.

Marking Scheme: Identification of Classes using Noun Phrase Approach: 1.5 Marks

Classes and Relations: 2.5 Marks

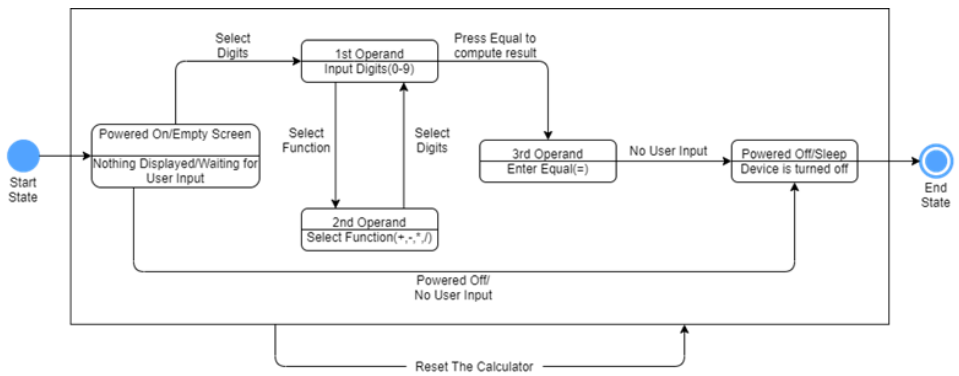




3. Draw a state transition diagram to model the simple calculator application. **3**
The calculator supports four basic operations +, -, *, /. To support these operations, the interface of the calculator is composed of 10 buttons with digits (0 to 9) and four buttons for the basic operations (+,-,*, /). A button 'C' to reset the display and '=' button to display the result along with 'ON' and 'OFF' buttons. The sequence to be followed for performing the basic operation is 'operand 1', 'operator selected', 'operand 2' and 'result display'. Other relevant transitions can be assumed.

Marking Scheme: Identification of correct states and Transitions: 1.5 Marks

State diagram showing states and transitions: 1.5 Marks



4. Define the term Software. Explain those characteristics of the software that makes it different from hardware. **3**

Software is: (1) instructions (computer programs) that when executed provide desired features, function, and performance; (2) data structures that enable the programs to adequately manipulate information, and (3) descriptive information in both hard copy and virtual forms that describes the operation and use of the programs.

0.5

• Software is a logical rather than a physical system element. Therefore, software has characteristics that are considerably different than those of hardware:

• Software is developed or engineered; it is not manufactured in the classical sense. **0.5**
Although some similarities exist between software development and hardware manufacturing, the two activities are fundamentally different. In both activities, high quality is achieved through good design, but the manufacturing phase for hardware can introduce quality problems that are nonexistent (or easily corrected) for software. Both activities are dependent on people, but the relationship between people applied and work accomplished is entirely different. Both activities require the construction of a "product," but the approaches are different. Software costs are concentrated in engineering. This means that software projects cannot be managed as if they were manufacturing projects.

• Software doesn't "wear out.": Hardware exhibits relatively high failure rates **0.5**
early in its life (these failures are often attributable to design or manufacturing defects); defects are corrected and the failure rate drops to a steady-state level (hopefully, quite low) for some period of time. As time passes, however, the failure rate rises again as hardware components suffer from the cumulative effects of dust, vibration, abuse, temperature extremes, and many other environmental maladies. Stated simply, the hardware begins to wear out. Software is not susceptible to the environmental maladies that cause hardware to wear out. Undiscovered defects will cause high failure rates early in the life of a program. So, software doesn't wear out. But it does deteriorate.

• When a hardware component wears out, it is replaced by a spare part. There are no software spare parts. Every software failure indicates an error in design or in the **0.5**
process through which design was translated into machine executable code. Therefore, the software maintenance tasks that accommodate requests for change involve considerably more complexity than hardware maintenance.

• Although the industry is moving toward component-based construction, most software continues to be custom built. Hardware manufacturing uses extensive reusable **0.5**
standard components. These reusable components have been created so that the engineer can concentrate on the truly innovative elements of a design, that is, the parts of the design that represent something new. In the hardware world, component reuse is a natural part of the engineering process. In the software world, it is something that has only begun to be achieved on a broad scale. A software component should be designed and implemented so that it can be reused in many different programs. Modern reusable components encapsulate both data and the processing that is applied to the data, enabling the software engineer to create new applications from reusable parts. For example, today's interactive user interfaces are built with reusable components that enable the creation of graphics windows, pull-down menus, and a wide variety of interaction mechanisms. The data structures and processing detail required to build the interface are contained within a library of reusable components for interface construction.

0.5

5. The cruise control of a car system is to maintain the speed set by the driver. The driver turns on cruise control in the car when a desired speed is reached. The desired speed must be always maintained. Slopes on the road, wind resistance, etc need to be accounted for, while keeping the speed constant. This is a simple open loop system with an input that goes into a controller. The control signal is processed as the output. Identify the architecture pattern the car cruise control system is modelled on with justification for the same. **3**

Identification of Pipe and filter pattern (.5M)

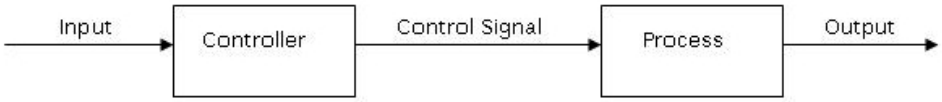


Illustration (.5M)
Two advantages (1M)
Two Disadantages(1M)

6. Design the test cases for the following code snippet using path testing. You are expected to follow the following steps to design the effective test case which have the high probability of revealing defects. **4**

1.Draw the CFG (Control Flow Graph) - 1 Mark
2.Find the Cyclomatic Complexity using three methods. - 1 Mark
3.Identify the independent paths (Basic Path Set) - 1 Mark
4.Derive test cases - 1 Mark

```
int main()
{
        int age;
        int cnt_baby=0,cnt_school=0,cnt_adult=0;
        int count=0;

        while(count<15)
        {
                printf("Enter age of person [%d]: ",count+1);
                scanf("%d",&age);

                if(age>=0 && age<=5)
                        cnt_baby++;
                else if(age>=6 && age<=17)
                        cnt_school++;
                else
                        cnt_adult++;


                count++;
        }

        printf("Baby age: %d\n",cnt_baby);
```

```
    printf("School age: %d\n",cnt_school);
    printf("Adult age: %d\n",cnt_adult);

    return 0;
}
```