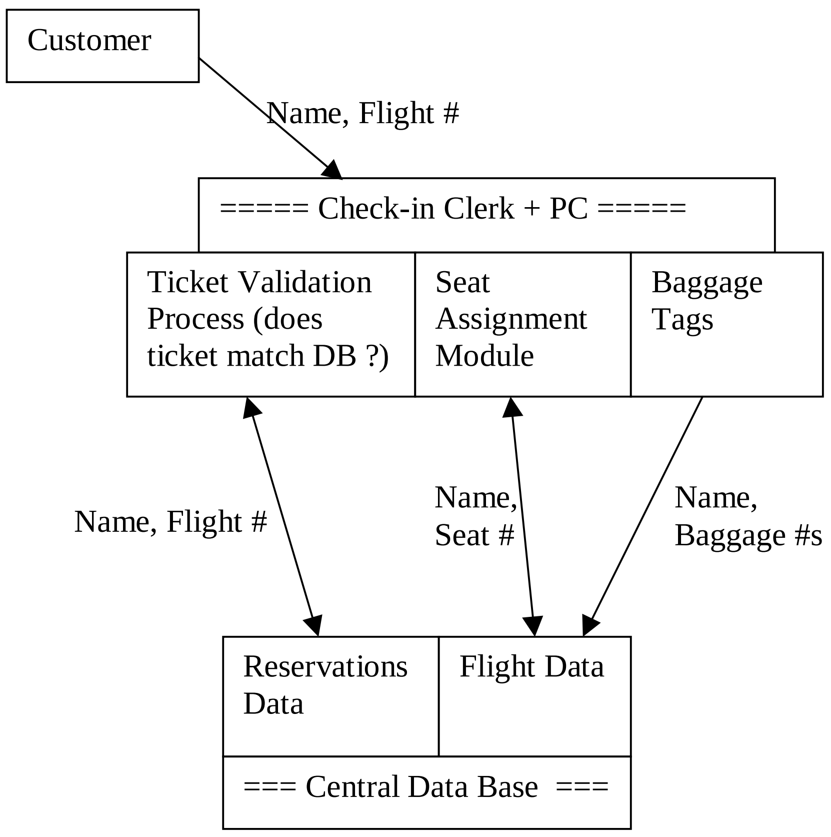
***** These answers are for the older version of these review questions, so some are incorrect *****
*****   especially the algorithms which are written in PURE instead of Java *****

QUESTIONS: Airplane Reservations

- Describe three significantly different **types of data** in this system.
  * Customer names, flight numbers, prices
- Describe three significantly different **users** of this system.
  * Travel agent, customer, airline companies
- For one of the **types of data**, estimate the **total quantity of input** required per week.
  * Customer names : if the airline has 20 flights per day, each with 100 customers, that would be 20*100*7 = 14000 names per week. If the name is in a 30 character field, that is 420 KB per week
- Describe two uses of a **computer terminal** by a travel agent in this system.
  * (1) Getting flight information about times, dates, and prices
  * (2) reserving a ticket for a specific customer on a specific flight
- Tickets can be printed at the travel agency. What **output device** is required?
  * printed output onto a ticket, which is a specially formatted form
  Does it have any special requirements other than normal functions?
  * The data must be printed into the boxes on the preprinted ticket.
- Describe a **processing feature** in this system which is **automated** and could probably not be done **manually.**
  ***** booking reservations could be done manually
- If a customer is to be **reminded automatically** of an upcoming flight, describe the **processing** required for this reminder, including **accessing stored data** and **producing output.**
  * We could write a rough algorithm for this
```
   search through all future reservations
  find all reservations for the date 1 week from today
  for each reservation
    get the corresponding name and phone/addreess/email (choose)
    notify that person
  next reservation
It may be useful to identify "family" reservations, and only send a
single reminder to the whole family.  This could be done be denoting
a customer as "notifiable" or "not-notifiable"
```
- A **flight attendant** needs to **input** data into this system – what data? Under what circumstances?
  * When the plane is boarding, the flight attendant must check that all the correct customers
  * board the plane. This could be done with a printed list of reservations, then checking off
  * whether each seat is full. It could also be done in a portable computer. The portable computer
  * has the advantage that it is easier and quicker to send this data back into the main system.
  * The data must be compared to the actual reservations very quickly, because take-off will
  * be delayed if it does not match.
- Explain what a **data-flow-diagram** is, and draw one for the process of **checking-in** at the airport.
  * A data-flow diagram has a box for each input module, each output module, each storage module,
  * and each processing (e.g. sorting) module. These are connected by arrows which show what
  * data flows between the various modules.   Module boxes can represent hardware or software or both.


Customer
Name, Flight #
===== Check-in Clerk + PC =====
Ticket Validation Process (does ticket match DB ?)
Seat Assignment Module
Baggage Tags
Name, Flight #
Name, Seat #
Name, Baggage #s
Reservations Data
Flight Data
=== Central Data Base ===

STAGES in system development:
The **investigation/analysis** stage should produce a **feasibility study** which involves:

- **ANALYSTS (and USERS)**
- User's goals and requests
- Stories from users, describing problems and needs
- Sample data
- Possible hardware choices (perhaps several alternatives)

-- Explain two reasons why it is desirable to have **different people** involved in the different stages.
* (1) The various people have different areas of expertise – it is bad if they are working outside this area,
*   and would slow things down and perhaps result in poor decisions.
* (2) Some decisions in one area may be **reversed** by other people in other areas – this is good.
-- In which stage does **module-level testing** occur?
* In the programming (implementation) stage.
-- In which two stages would a **questionnaire** be usefule?
* Investigation/anslysis and Maintenance/review.

1.3 Software Design

* OMR = Optical Mark Recognition – pencil bubble sheets commonly used for multiple choice tests
* GUI = Graphical User Interface, as opposed to a Command Line (Text) interface
* (1) store it in a string as "05/23/02"
  (2) store in an array, as Date[1] = 5 Date[2] = 23 Date[3] = 2002
  (3) Store it in a long integer as 20020523.
* 1234567890987654321 = it's too big - overflow
* 1 e –123456789 = too small = underflow
* All the ASCII codes below 32 represent printer control codes, not characters. Zero (0) is "never" used.
* A list of names. If there are too many names, it might **overflow** the size (dimension) of the array.
* Names(1000) as string, Ages(1000) as integer
   The Name of one person is stored in the Names array, and his/her age is store in the same position in Ages.
* If the file is too large, it might not fit in the RAM, so it won't fit in an array.
* To represent a chess-board, to represent seating in a theatre

A common **algorithm** is the **bubble-sort.**
***** If the array is already sorted, the bubble-sort will make no swaps in the first pass, and could then stop.
* Use a **parameter** for the name of the array (e.g. LIST), then call SORT(Names), SORT(Cities), ...
* It has nested loops, so 1000 items need 1000*1000 iterations = 1 million
   8000 items needs 8000*8000 = 64 million iterations, so it takes 64 times longer = 640 seconds = 10.7 min
* A selection sort will be faster on scrambled up data, because each pass is shorter than the previous one.
* If the file is already sorted, the bubble sort **could** quit after one pass, whereas a selection sort
    always requires the same number of iterations – it cannot stop early.

Another common algorithm is a **sequential search.**
***** If it finds the name it is looking for, or in a sorted list if the target name is smaller than the current item.
* Faster in an array because disk-access is much slower than memory access
* Binary search – start in the middle .....etc
* Takes 8 times longer (single loop), so 80 sec.

1.4 Constructs

* scope = where in the program can the variable be used
* global variables have larger scope
* a parameter is very similar to a local variable
* A **record** like:

```
class Student
{  String name;
   int age;
   String[] address = new String[4];
}
```

* Something like (x>4) , (name = "Bob"), using the operations > , < , =
  The result of a comparison is a boolean value **true/false**
* Something like (x < 5) and not eof(datafile), using boolean operations AND, OR, NOT, XOR
  The result of a boolean expression is a boolean value **true/false**
* In PURE, the **operator precedence** is:

  1 : brackets
  2 : arithmetic operators, * / before + -
  3 : comparisons, < > = # in order from left to right
  4 : boolean : not first, then and, then or/xor

* A **record** can contain many different types of data in different ___fields______________ .
* a **selection construct = if..then..**
* **looping constructs** : **for..enfor while..endwhile repeat..until..**
* **iteration construct** : the same as a **looping construct**
* **sentinel values** are strange data values like "XXX" which mark the end of an array (or a file)
* **flag :** a **boolean** variable to keep track of whether something has happened or not,
  for example FOUND in a search algorithm.
* **by-reference :** any changes to the parameter inside the procedure are copied back to the calling routine
  by-value: input only – changes are not copied back to the calling routine.

Long Question

```
function COUNT(val LIST string array)
   result integer
   declare C,POS integer
   POS <-- 1
   C <-- 0
   while ( LIST[POS] <> "XXX" ) do
      C <-- C + 1
      POS <-- POS + 1
   endwhile
   return C
endfunction

procedure REVERSE(val LIST string array)
   declare X, START integer
   START <-- COUNT(LIST)-1
   for X <-- START downto 1 do
      output LIST[X]
   endfor
endprocedure
```

1.4 Errors and Testing

```
function ADDUP(val N integer)
   result integer
   declare X, SUM integer
   SUM <-- 0
   for X <-- 1 to N
      SUM <-- SUM + X
   endfor
   return SUM
endfunction
```

***** incorrect answer = logic error
* missing letter = syntax error
* no, the result would be zero, no problem
* The compiler detects a syntax error when something is written incorrectly.
But run-timer errors result from bad data or some other situation which the compiler cannot predict.
* ADDUP(10) , expected answer = 55
ADDUP(0) , expected answer 0
ADDUP(1), expected answer 1
* A very large value 123465691261928034619023 might cause an overflow.
prevent this with an if..then..:

```
  If N < 1000000 then
       do the procedure
     else
       output error message
       return -1
   endif
```

* Instant error messages when a syntax error is typed
DEBUG (trace) mode showing which statements are being executed
Syntax highlighting (colors)
Break at run-time error showing offending statement
* Print the program listing on paper, take a pencil, and READ the program,
trying to "run" the program step by step with sample data

*** infinite loop =** run-time error **and** logic error

Longer Question

Consider the following algorithm:

```
main
   output ROUNDED(19.9559);
endmain

function ROUNDING(val NUM integer)
   result integer
   return truncate(NUM + 0.5)
endfunction
```

* syntax error : ROUNDED and ROUNDING
***** type mismatch caused by parameters : val NUM real would be better.
* output = 20
* wrong output = Logic error

1.5 Documentation

***** Internal docs talk about program code and data structures (how the program was written)
while user docs talk about how to USE the program
Internal docs usually appear inside the program listing
while user docs are contained in a separate book
Internal docs are used when the program needs to be fixed or changed
while user docs talk about how to install and maintain the program
***** A **technical writer** should write them. This person should have some technical understanding
and significant level of understanding of the application area (user perspective).
This person serves as an interface between the programmers (developers) and the users.

NEW Language

* the **position** of the **remote** identifier is a **syntax** issue
* the choice of the word **remote** or **extern** is a **semantics** issue
*** $ sign** is both syntax and semantics issue
* Interpretter must translate each command over and over again,
and will be making long-distance connections over and over again, which
might be worse than with a compiler
* intended for both LAN and WAN, as there will also be members of the programming team
locally in Germany
* HTML is **not** a programming language
* The CASE Tool should run on the client, because it must have access to the Indian programmer's
part of the project, which would not work from the server side.

===========================

2.2 Architecture

They are now matched here:

Device	**Application**
OCR	change a printed page back into a word-processing document
sensor	respond to movement or light signals
printer	create bar-coded sticky labels for library books
MICR	print an ID number on a bank chequie
mouse	select from a drop-down menu on a PC
scanner	copy a photograph into the computer
modem	transmit packets of data from one computer to another
graphics tablet	draw accurate pictures with a stylus
cell phone	convert analog sound to digital data and transmit it from various locations
LCD panel	display data in a portable computer
keyboard	give commands in a command-line interface
digital camera	capture pictures which can be displayed over the Internet
speech recognition	give commands to a computer when your hands are busy
sound card	input music
plotter	draw accurate architectural diagrams, without jaggies
touch-screen	start a program in a pocket-organizer

2.2.2 Chips

The ALU and CU are two parts of the ___CPU______. The ALU performs _arithmetic____ and

___logic______ operations. The ___CU________ controls communications across the data bus.

The data bus contains 32 (or maybe 64) __parallel_ circuits,

each of which can carry one ____bit_____ of data.

The CPU uses the data bus to fetch and store data in the _______RAM__________.

But to speed up operations, the data bus does not connect directly between the CPU and the RAM.

The CPU uses the ________cache__________ for temporary storage, and **it** is connected to the RAM.

The speed of the CPU might be 1 ___GIGA______ Herz. This means __1 billion______ cycles per second.

If the data bus is 64 ___bits__ wide, and runs at 266 ___megaherz__,

then it can transfer data at a speed of _8 bytes * 266 MegaHz = _2128__ MegaBytes / Sec.

This is the same as __2.1____ GigaBytes per second.

Earlier computers did not run at such high speeds. Running a CPU at a higher speed generates a lot of __heat__. Cooling fans can only help a certain amount. Super computers use liquid nitrogen to cool the circuits and chips, which can then run at very high frequencies without damage. In PCs, the solution is to make the circuits in the chips __smaller____, reducing the size from 0.25 microns to 0.13 microns, and then also reducing the voltage from 3 V to 1.5 V. Using less energy produces less heat, so the chip can run at a faster frequency without overheating. It is also possible to place more ___transistors_ on the chip in the same space.

A Pentium chip contains over 25 ___million____ transistors.

In RAM, a transistor represents one __bit____. 8 __bits___ makes 1 ___byte____

Modern PCs contain 128 ___megabytes___ of RAM, or more. A hard-disk might store 40 ___GigaBytes_ of data. The hard-disk can store _40 * 1024 / 1.44 = 28,444 _ times as much data as the RAM.

When a program runs, it must
be copied into the ___RAM________. If several programs are running at once (__Multitasking_____), the RAM could get full. Then the CPU must **swap** some of the memory out onto the ____hard disk____, to free up more memory for other programs. This process is called ___virtual______ memory, because the computer seems to have more memory than the actual physical RAM.

2.2 Storage

T = Tera = 2 ^ __40____ 1 Terabyte is equal to __1 million_ MegaBytes.

G = Giga = 2^ ____30__ A 40 GigaByte hard disk can hold as much as _28,444__ floppy diskettes.

M = Mega = 2^ __20____ Using a 56 K modem, the fastest possible download of 1 MB is _142___ sec.

K = Kilo = 2^___10____ A hard-disk sector containing 512 bytes is the same as __0.5 __ KB.

A 32 bit address bus can carry addresses between 0 and ____4 billion__ and address a maximum

of _____4 GigaBytes_____ RAM.

A _____serial__________ interface has only 1 data wire, and transmits 1 bit after the next.

If it runs at 240 KHz, it can trasmit _240 K / 8 = 30 KB per second.

A __parallel____ interface uses 8 data wires, and thus can transmit an entire byte at one time.

================================

* Kilometers are a base-10 measurement, and 1000 is a power of 10 (10^3).
KiloBytes are counted in binary, and 1024 is a power of 2 (2^10)
* In sequential access, the data must all be read in order, from the beginning to the end.
In direct access, it is possible to "jump" to any position in the file without reading the previous data.
* Backups are commonly done on tape-drives, which can only be used in sequential access mode
Tape is used because it is cheap and can be removed and stored somewhere else for safety,
and the speed of the backup process is not an important issue.
* Smaller circuits squeeze more circuit elements into the same space. The manufacturing cost
depends more on the physical size of the chip – if that stays the same, the cost stays the same.
* The **pits (holes)** on the DVD disk are smaller and packed closer together than the CD-ROM

===================================
* to "jump" into the middle of the file, the computer must **calculate** the number of bytes between the
beginning of the file and the desired record. This is only possible if all records are the same size (in bytes)
===================================

Long Question

A new **pocket PC** contains no disk drive. Instead it uses **flash memory**. This is a large amount of **RAM** (e.g. 64 MB) with a constant power supply, so the data never gets erased. A special design achieves this without using very much power, so batteries can last several weeks or months. But flash memory is expensive.

* The OS seldom changes, but is used very often, so the ROM is an efficient solution.
* Link cable to a PC, copy data to PC disk drive
* Screen blanking saves power – an important issue in portable computers
* The screen is far too small to display web-screens – typically something like 300x200 pixels.
* There is no keyboard, so speech-recognition is a very attractive possibility for data input.

2.3 Systems

* Windows is a multi-tasking GUI.
* The server does not need a GUI, but requires very high reliability and very efficient multi-tasking
* Multi-tasking
* A main-frame probably is a mult-user system, so multi-processors is a great advantage.
* Data Files, printers, disk storage space
* E-mail client (your PC) and the E-mail server (collects and transmits mail)
* Batch proceed through the entire process without user interaction
* Posting transactions (adding/subtracting money) to user account balances
* Verifying a PIN code for an ATM transaction
* Searching for overdue books and printing warning notices (done daily or weekly)
* Checking out a book
* Anything that happens in a library is not urgent – it can wait
* Air-traffic control **cannot** wait – it must give up-to-date information constantly
* intensive care = real-time

2.4 Data Representation

Analog/Digital
Which of the following involve DAC or ADC? Which do not?

- Storing data on a disk drive	**Digital**	- Playing CD music on a PC speaker	**DAC**
- Playing a phonograph record	**Analog**	- Measuring temperatures and storing them	**ADC**
- Using a phone-card in a	**Digital**	- Controlling an industrial robot	**DAC**
- Printing from a PC to a laser printer	**Digital**	- Audio sampling (what's that?)	**ADC**
- Photography with a digital camera	**ADC**	- Scanning and OCR	**ADC**

Binary
***** 24-bit color is called **true color.** How many different colors are there? 2^24 = 16 million
* One byte can contain any positive number between _0_____ and __255__.
* How many bits are needed to store the number 1,000,000 ? **20 bits**
* Write the following IP address as a 32-bit binary number: 179.123.10.99
The four numbers are individual bytes, and can be converted independentley:
10110011 01111011 00001010 01100011

* If the byte 10101100 is transmitted, what **parity bit** should be transmitted with it (even parity)? **0**
* Explain what a **checksum** is. Is it most appropriate with a byte, a packet, a file, or a disk drive?
Add up all the bytes, and send along the sum with the transmission.
Most appropriate for a packet.