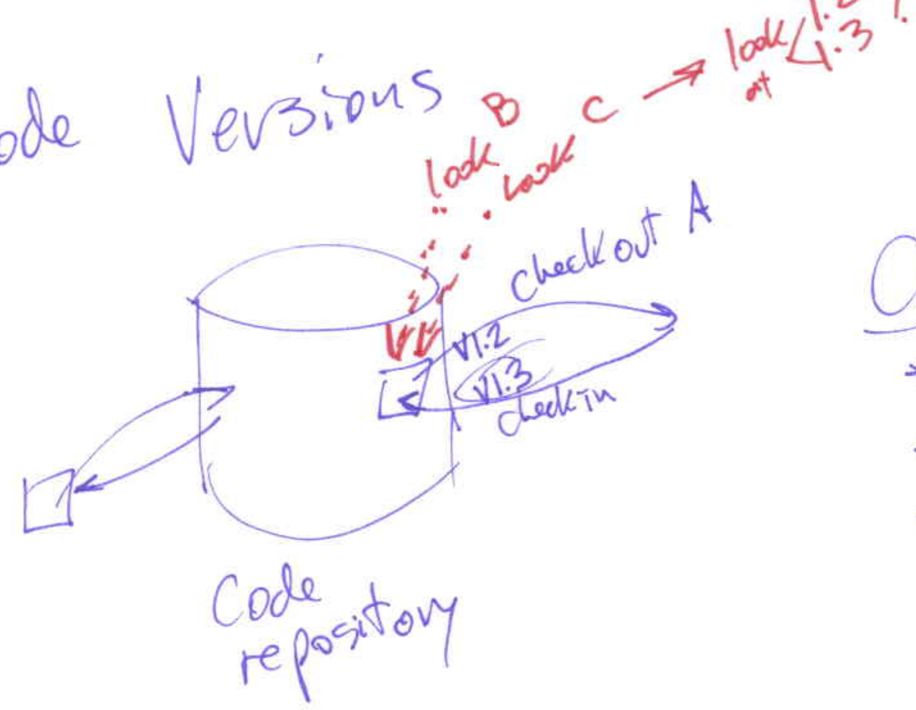
ECGR 4101/5101 10/21/13

Code Versions



Check Out

* Modify
* Reviewed
* Compiled w latest version of all files
* Tested
* Then → Check in

Time to make next version of code base:

changed $\boxed{20}$ module (of 1000)
2%

Use or "Roll back" to a previous version

Lab 4. Ver 1.9 worked ← copy it
Ver 1.10 Did not work

"It does not work <u>anymore</u>...."

ECGR 4101/5101

```
// Quiz 11 code

#define Q_SIZE (8)
typedef struct {
  unsigned char Data[Q_SIZE];
  unsigned int Head; // points to oldest data element
  unsigned int Tail; // points to next free space
  unsigned int Size; // quantity of elements in queue
} Q_T;
Q_T tx_q, rx_q;

void Q_Init(Q_T * q) {
  unsigned int i;
  for (i=0; i<Q_SIZE; i++)
    q->Data[i] = 0;  // to simplify our lives when debugging
  q->Head = 0;
  q->Tail = 0;
  q->Size = 0;
}
int Q_Empty(Q_T * q) {
  return q->Size == 0;
}
int Q_Full(Q_T * q) {
  return q->Size == Q_SIZE;
}

// Q_Enqueue - Called by a UART ISR - put a char on the queue
int Q_Enqueue(Q_T * q, unsigned char d) {
  if (!Q_Full(q)) { // What if queue is full?
    q->Data[q->Tail++] = d;
    q->Tail %= Q_SIZE;
    q->Size++;
    return 1; // success
  } else
    return 0; // failure
}
// Q_Dequeue-called by a consumer function-take a char from queue
unsigned char Q_Dequeue(Q_T * q) {
  unsigned char t=0;
  if (!Q_Empty(q)) {     // Must check to see if queue is empty 1st
    t = q->Data[q->Head];
    q->Data[q->Head++] = 0;    // to simplify debugging, clear
    q->Head %= Q_SIZE;
    q->Size--;
  }
  return t;
}
```

0	~~0~~ ~~41~~ ~~0~~ 45
1	~~0~~42
2	~~0~~30
3	~~0~~31
4	~~0~~32
5	~~0~~33
6	~~0~~3A
7	039

Head ~~0~~ 1
Tail ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~7~~ ~~8~~ 0 1
Size ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~2~~ ~~6~~ ~~7~~ 8

ISR x41
ISR x42
ISR x30
Main consume t=~~4~~41
ISR x31
ISR x32
ISR x33
ISR x34
ISR x39
ISR x45
ISR x41
MainConsume
MainConsume
MainConsume
ISR