

ITEM 9:

Never Call Virtual functn during constructon or Destruction.

Ex → class Transactn

{ public:

Transaction();

virtual void logTran (T const = 0);

};

Transaction::Transaction()

{ logTran(); }

class BuyTran : public Transaction

{ public:

void logTran () const;

};

BuyTran bt;

The version of logTran called is of Transaction & not BuyTran since base class parts of derived class objects are constructed before derived class parts.

Next if we have something like virtual functns called during base class constructn next down to derived class, then the data

members of those would not have been initialized yet, to avoid this make logTran as non-virtual function and in BuyTran make a static function so that there is no danger of accidentally referring to BuyTran object as yet uninitialized data member.

```

class Transaction
{
    Transaction(const string &logInfo);
    void logTran(const string &logInfo) const;
};

Transaction::Transaction(const string &logInfo)
{
    logTran(logInfo);
}

class BuyTran: public Transaction
{
public:
    BuyTransaction(para):
        Transaction(createLogString(para));
private:
    static createLogString(para);
};
  
```