Table C55 PSP2 Project Plan Summary

Student _____ Date _____
Program _____ Program # _____
Instructor _____ Language _____

Summary	**Plan**	**Actual**	**To Date**
LOC/Hour			
Planned Time			
Actual Time			
CPI(Cost-Performance Index)			(Planned/Actual)
% Reused			
% New Reused			
Test Defects/KLOC			
Total Defects/KLOC			
Yield %			

Program Size (LOC):	**Plan**	**Actual**	**To Date**
Base(B)	(Measured)	(Measured)	
Deleted (D)	(Estimated)	(Counted)	
Modified (M)	(Estimated)	(Counted)	
Added (A)	(N-M)	(T-B+D-R)	
Reused (R)	(Estimated)	(Counted)	
Total New & Changed (N)	(Estimated)	(A+M)	
Total LOC (T)	(N+B-M-D+R)	(Measured)	
Total New Reused			
Upper Prediction Interval (70%)			
Lower Prediction Interval (70%)			

Time in Phase (min.)	**Plan**	**Actual**	**To Date**	**To Date %**
Planning				
Design				
Design review				
Code				
Code review				
Compile				
Test				
Postmortem				
Total				
Total Time UPI (70%)				
Total Time LPI (70%)				

(continued)

Table C55 PSP2 Project Plan Summary (continued)

Student		Date	
Program		Program #	
Instructor		Language	

Defects Injected	***Plan***	**Actual**	**To Date**	**To Date %**
Planning				
Design				
Design review				
Code				
Code review				
Compile				
Test				
Total Development				

Defects Removed	***Plan***	**Actual**	**To Date**	**To Date %**
Planning				
Design				
Design review				
Code				
Code review				
Compile				
Test				
Total Development				
After Development				

Defect Removal Efficiency	***Plan***	***Actual***	***To Date***
Defects/Hour - Design review			
Defects/Hour - Code review			
Defects/Hour - Compile			
Defects/Hour - Test			
DRL(DLDR/UT)			
DRL(CodeReview/UT)			
DRL(Compile/UT)			

Table C57 C++ PSP2 Design Review Checklist

PROGRAM NAME AND #:

Purpose	To guide you in conducting an effective design review				
General	As you complete each review step, check that item in the box to the right. Complete the checklist for one program unit before you start to review the next.				
Complete	Ensure that the requirements, specifications, and high-level design are completely covered by the design: - all specified outputs are produced - all needed inputs are furnished - all required includes are stated				
Logic	Verify that program sequencing is proper: - that stacks, lists, etc. are in the proper order - that recursion unwinds properly Verify that all loops are properly initiated, incremented, and terminated				
Special Cases	Check all special cases: - empty, full, minimum, maximum, negative, zero - out of limits, overflow, underflow - ensure "impossible" conditions are absolutely impossible - handle all incorrect input conditions				
Functional use	Verify that all functions, procedures, or objects are fully understood and properly used Verify that all externally referenced abstractions are precisely defined				
Names	Verify that: - all special names and types are clear or specifically defined - the scopes of all variables and parameters are self-evident or defined - all named objects are used within their declared scopes				
Standards	Review the design for conformance to all applicable design standards				

Table C58 C++ Code Review Checklist

PROGRAM NAME AND #:

Purpose	To guide you in conducting an effective code review.				
General	As you complete each review step, check that item in the box to the right. Complete the checklist for one program unit before you start to review the next.				
Complete	Verify that the code covers all the design.				
Includes	Verify that includes are complete				
Initialization	Check variable and parameter initialization: - at program initiation - at start of every loop - at function/procedure entry				
Calls	Check function call formats: - pointers - parameters - use of '&'				
Names	Check name spelling and use: - is it consistent? - is it within declared scope? - do all structures and classes use '.' reference?				
Strings	Check that all strings are - identified by pointers and - terminated in NULL.				
Pointers	Check that - pointers are initialized NULL - pointers are deleted only after new, and - new pointers are always deleted after use.				
Output Format	Check the output format: - line stepping is proper - spacing is proper				
{} Pairs	Ensure that the {} are proper and matched				
Logic Operators	Verify the proper use of ==, =, ||, and so on. Check every logic function for proper ().				
Line by Line Check	Check every LOC for - instruction syntax and - proper punctuation.				
Standards	Ensure that the code conforms to the coding standards.				
File Open and Close	Verify that all files are - properly declared, - opened, and - closed.				

Table C39 Size Estimating Template

Student _____ Date _____

Instructor _____ Program # _____

BASE PROGRAM	LOC
BASE SIZE (B) => => => => => => => => => =>	
LOC DELETED (D) => => => => => => => => => =>	
LOC MODIFIED (M) => => => => => => => => => =>	

PROJECTED LOC

BASE ADDITIONS:	TYPE	METHODS	REL. SIZE	LOC
TOTAL BASE ADDITIONS (BA) => => => => => =>				

NEW OBJECTS:	TYPE[1]	METHODS	REL. SIZE	LOC (NewReuse*)
TOTAL NEW OBJECTS (NO) => => => => => => =>				

REUSED PROGRAMS	LOC
REUSED TOTAL (R) => => => => => => => => => =>	

Projected LOC:	P = BA+NO	
Regression Parameter:	β_0	
Regression Parameter:	β_1	
Estimated New and Changed LOC:	N = β_0 + β_1*(P+M)	
Estimated Total LOC:	T = N + B - D - M + R	
Estimated Total New Reused (sum of * LOC):		
Prediction Range:	Range	
Upper Prediction Interval:	UPI = N + Range	
Lower Prediction Interval:	LPI = N - Range	
Prediction Interval Percent:		

[1] L-Logic, I-I/O, C-Calculation, T-Text, D-Data, S-Set-up