

Copyright 2018 Carnegie Mellon University. All rights reserved.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

**NO WARRANTY. THIS MATERIAL IS FURNISHED ON AN “AS-IS” BASIS WITH NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, ANY WARRANTY WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT, OR THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.**

[Distribution Statement A] This material has been approved for public release and unlimited distribution. The United States Government has Unlimited Rights in this material as defined by DFARS 252.227-7013.

The text and illustrations in this material are licensed by Carnegie Mellon University under a [Creative Commons Attribution 4.0 International License](#).

The Creative Commons license does not extend to logos, trade marks, or service marks of Carnegie Mellon University.



## PSP Scripts, Forms, Templates, and Standards

This document contains all of the PSP process materials, as listed in the following table.

### PSP Cross-reference

Heading	Page Numbers						
Process Version		PSP0	PSP0.1	PSP1	PSP1.1	PSP2	PSP2.1
<b>Process Scripts and Summaries</b>							
PSP Process Script		2	13	23	37	48	60
PSP Planning Script		3	14	24	38	49	61
PSP Development Script		4	15	25	39	50	62
PSP Design Review Script						51	63
PSP Code Review Script						52	X
PSP Postmortem Script		5	16	26	40	53	64
Project Plan Summary and Instructions	6, 7	17, 18	27, 28	41, 43	54, 56	65, 67	
PROBE Estimating Script	29			X	X	X	X
<b>Forms, Templates, Standards, and Instructions</b>							
Time Recording Log	8, 9	X	X	X	X	X	X
Defect Recording Log	10, 11	X	X	X	X	X	X
Defect Type Standard	12	X	X	X	X	X	X
PIP	19, 20		X	X	X	X	X
Coding Standard	21		X	X	X	X	X
Test Report Template	31, 32			X	X	X	X
Size Estimating Template	33, 35			X	X	X	X
Task Planning Template	44, 45				X	X	X
Schedule Planning Template	46, 47				X	X	X
Design Review Checklist						58	69
Code Review Checklist	59					X	X
Use Case Specification Template	70, 71						X
Functional Specification Template	72, 73						X
State Specification Template	74, 75						X
Logic Specification Template	76, 77						X
Expanded Defect Type Standard	78						
Process Development Process Script (PDP)	79						
PSP3.0 Development Script	80						
Prototype Experimental Process Script (PEP)	81						
Product Maintenance Process Script (PMP)	82						

**PSP0 Process Script**

<b>Purpose</b>	To guide the development of module-level programs	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Problem description</li> <li>- PSP0 Project Plan Summary form</li> <li>- Time and Defect Recording logs</li> <li>- Defect Type standard</li> <li>- Stopwatch (optional)</li> </ul>	
<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Planning	<ul style="list-style-type: none"> <li>- Produce or obtain a requirements statement.</li> <li>- Estimate the required development time.</li> <li>- Enter the plan data in the Project Plan Summary form.</li> <li>- Complete the Time Recording log.</li> </ul>
2	Development	<ul style="list-style-type: none"> <li>- Design the program.</li> <li>- Implement the design.</li> <li>- Compile the program, and fix and log all defects found.</li> <li>- Test the program, and fix and log all defects found.</li> <li>- Complete the Time Recording log.</li> </ul>
3	Postmortem	Complete the Project Plan Summary form with actual time, defect, and size data.
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- A thoroughly tested program</li> <li>- Completed Project Plan Summary form with estimated and actual data</li> <li>- Completed Time and Defect Recording logs</li> </ul>	

**PSP0 Planning Script**

<b>Purpose</b>	To guide the PSP planning process
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Problem description</li> <li>- Project Plan Summary form</li> <li>- Time Recording log</li> </ul>

<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Program Requirements	<ul style="list-style-type: none"> <li>- Produce or obtain a requirements statement for the program.</li> <li>- Ensure that the requirements statement is clear and unambiguous.</li> <li>- Resolve any questions.</li> </ul>
2	Resource Estimate	Make your best estimate of the time required to develop this program.

<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- Documented requirements statement</li> <li>- Completed Project Plan Summary form with estimated development time data</li> <li>- Completed Time Recording log</li> </ul>
----------------------	---

**PSP0 Development Script**

<b>Purpose</b>	To guide the development of small programs	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Requirements statement</li> <li>- Project Plan Summary form with estimated program development time</li> <li>- Time and Defect Recording logs</li> <li>- Defect Type standard</li> </ul>	
<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Design	<ul style="list-style-type: none"> <li>- Review the requirements and produce a design to meet them.</li> <li>- Record in the Defect Recording log any requirements defects found.</li> <li>- Record time in the Time Recording log.</li> </ul>
2	Code	<ul style="list-style-type: none"> <li>- Implement the design.</li> <li>- Record in the Defect Recording log any requirements or design defects found.</li> <li>- Record time in the Time Recording log.</li> </ul>
3	Compile	<ul style="list-style-type: none"> <li>- Compile the program until there are no compile errors.</li> <li>- Fix all defects found.</li> <li>- Record defects in the Defect Recording log.</li> <li>- Record time in the Time Recording log.</li> </ul>
4	Test	<ul style="list-style-type: none"> <li>- Test until all tests run without error.</li> <li>- Fix all defects found.</li> <li>- Record defects in the Defect Recording log.</li> <li>- Record time in the Time Recording log.</li> </ul>
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- A thoroughly tested program</li> <li>- Completed Time and Defect Recording logs</li> </ul>	

**PSP0 Postmortem Script**

<b>Purpose</b>	To guide the PSP postmortem process	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Problem description and requirements statement</li> <li>- Project Plan Summary form with development time data</li> <li>- Completed Time and Defect Recording logs</li> <li>- A tested and running program</li> </ul>	
<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Defect Recording	<ul style="list-style-type: none"> <li>- Review the Project Plan Summary to verify that all of the defects found in each phase were recorded.</li> <li>- Using your best recollection, record any omitted defects.</li> </ul>
2	Defect Data Consistency	<ul style="list-style-type: none"> <li>- Check that the data on every defect in the Defect Recording log are accurate and complete.</li> <li>- Verify that the numbers of defects injected and removed per phase are reasonable and correct.</li> <li>- Using your best recollection, correct any missing or incorrect defect data.</li> </ul>
3	Time	<ul style="list-style-type: none"> <li>- Review the completed Time Recording log for errors or omissions.</li> <li>- Using your best recollection, correct any missing or incomplete time data.</li> </ul>
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- A thoroughly tested program</li> <li>- Completed Project Plan Summary form</li> <li>- Completed Time and Defect Recording logs</li> </ul>	

## PSP0 Project Plan Summary

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

<b>Time in Phase (min.)</b>	<b>Plan</b>	<b>Actual</b>	<b>To Date</b>	<b>To Date %</b>
Planning	_____	_____	_____	_____
Design	_____	_____	_____	_____
Code	_____	_____	_____	_____
Compile	_____	_____	_____	_____
Test	_____	_____	_____	_____
Postmortem	_____	_____	_____	_____
Total	_____	_____	_____	_____

<b>Defects Injected</b>	<b>Actual</b>	<b>To Date</b>	<b>To Date %</b>
Planning	_____	_____	_____
Design	_____	_____	_____
Code	_____	_____	_____
Compile	_____	_____	_____
Test	_____	_____	_____
Total Development	_____	_____	_____

<b>Defects Removed</b>	<b>Actual</b>	<b>To Date</b>	<b>To Date %</b>
Planning	_____	_____	_____
Design	_____	_____	_____
Code	_____	_____	_____
Compile	_____	_____	_____
Test	_____	_____	_____
Total Development	_____	_____	_____
After Development	_____	_____	_____

### PSP0 Plan Summary Instructions

<b>Purpose</b>	To hold the plan and actual data for programs or program parts
<b>General</b>	“To Date” is the total actual to-date values for all products developed.
<b>Header</b>	<ul style="list-style-type: none"> <li>- Enter your name and the date.</li> <li>- Enter the program name and number.</li> <li>- Enter the instructor’s name and the programming language you are using.</li> </ul>
<b>Time in Phase</b>	<ul style="list-style-type: none"> <li>- Enter the estimated total development time.</li> <li>- Enter the actual time by phase and the total time.</li> <li>- To Date: Enter the sum of the actual times for this program plus the to-date times from the most recently developed program.</li> <li>- To Date %: Enter the percentage of to-date time in each phase.</li> </ul>
<b>Defects Injected</b>	<ul style="list-style-type: none"> <li>- Enter the actual defects by phase and the total actual defects.</li> <li>- To Date: Enter the sum of the actual defects injected by phase and the to-date values for the most recent previously developed program.</li> <li>- To Date %: Enter the percentage of the to-date defects injected by phase.</li> </ul>
<b>Defects Removed</b>	<ul style="list-style-type: none"> <li>- To Date: Enter the actual defects removed by phase plus the to-date values for the most recent previously developed program.</li> <li>- To Date %: Enter the percentage of the to-date defects removed by phase.</li> <li>- After development, record any defects subsequently found during program testing, use, reuse, or modification.</li> </ul>

# PSP Time Recording Log

Student \_\_\_\_\_ Date \_\_\_\_\_  
Program \_\_\_\_\_ Program # \_\_\_\_\_  
Instructor \_\_\_\_\_ Language \_\_\_\_\_

### Time Recording Log Instructions

<b>Purpose</b>	<ul style="list-style-type: none"> <li>- Use this form to record the time you spend on each project activity.</li> <li>- For the PSP, phases often have only one activity; larger projects usually have multiple activities in a single process phase.</li> <li>- These data are used to complete the Project Plan Summary.</li> <li>- Keep separate logs for each program.</li> </ul>
<b>General</b>	<ul style="list-style-type: none"> <li>- Record all of the time you spend on the project.</li> <li>- Record the time in minutes.</li> <li>- Be as accurate as possible.</li> <li>- If you need additional space, use another copy of the form.</li> <li>- If you forget to record the starting, stopping, or interruption time for an activity, promptly enter your best estimate.</li> </ul>
<b>Header</b>	<ul style="list-style-type: none"> <li>- Enter your name and the date.</li> <li>- Enter the program name and number.</li> <li>- Enter the instructor's name and the programming language you are using.</li> </ul>
<b>Project</b>	Enter the program name or number.
<b>Phase</b>	Enter the name of the phase for the activity you worked on, e.g. Planning, Design, Test.
<b>Start Date and Time</b>	Enter the date and time when you start working on a process activity.
<b>Interruption Time</b>	<ul style="list-style-type: none"> <li>- Record any interruption time that was not spent on the process activity.</li> <li>- If you have several interruptions, enter their total time.</li> <li>- You may enter the reason for the interrupt in comments.</li> </ul>
<b>Stop Date and Time</b>	Enter the date and time when you stop working on that process activity.
<b>Delta Time</b>	Enter the clock time you actually spent working on the process activity, less the interruption time.
<b>Comments</b>	Enter any other pertinent comments that might later remind you of any unusual circumstances regarding this activity.

**PSP Defect Recording Log**

Defect Types	
10 Documentation	60 Checking
20 Syntax	70 Data
30 Build, Package	80 Function
40 Assignment	90 System
50 Interface	100 Environment

Student \_\_\_\_\_ Date \_\_\_\_\_  
 Program \_\_\_\_\_ Program # \_\_\_\_\_  
 Instructor \_\_\_\_\_ Language \_\_\_\_\_

Project	Date	Number	Type	Inject	Remove	Fix Time	Fix Ref.
<input type="text"/>							

Description: \_\_\_\_\_

---

Project	Date	Number	Type	Inject	Remove	Fix Time	Fix Ref.
<input type="text"/>							

Description: \_\_\_\_\_

---

Project	Date	Number	Type	Inject	Remove	Fix Time	Fix Ref.
<input type="text"/>							

Description: \_\_\_\_\_

---

Project	Date	Number	Type	Inject	Remove	Fix Time	Fix Ref.
<input type="text"/>							

Description: \_\_\_\_\_

---

Project	Date	Number	Type	Inject	Remove	Fix Time	Fix Ref.
<input type="text"/>							

Description: \_\_\_\_\_

---

Project	Date	Number	Type	Inject	Remove	Fix Time	Fix Ref.
<input type="text"/>							

Description: \_\_\_\_\_

---

Project	Date	Number	Type	Inject	Remove	Fix Time	Fix Ref.
<input type="text"/>							

Description: \_\_\_\_\_

---

Project	Date	Number	Type	Inject	Remove	Fix Time	Fix Ref.
<input type="text"/>							

Description: \_\_\_\_\_

---

### PSP Defect Recording Log Instructions

<b>Purpose</b>	<ul style="list-style-type: none"> <li>- Use this form to hold data on the defects that you find and correct.</li> <li>- These data are used to complete the Project Plan Summary form.</li> </ul>
<b>General</b>	<ul style="list-style-type: none"> <li>- Record each defect separately and completely.</li> <li>- If you need additional space, use another copy of the form.</li> </ul>
<b>Header</b>	<ul style="list-style-type: none"> <li>- Enter your name and the date.</li> <li>- Enter the program name and number.</li> <li>- Enter the instructor's name and the programming language you are using.</li> </ul>
<b>Project</b>	<ul style="list-style-type: none"> <li>- Give each program a different name or number.</li> <li>- For example, record test program defects against the test program.</li> </ul>
<b>Date</b>	Enter the date on which you found the defect.
<b>Number</b>	<ul style="list-style-type: none"> <li>- Enter the defect number.</li> <li>- For each program or module, use a sequential number starting with 1 (or 001, etc.).</li> </ul>
<b>Type</b>	<ul style="list-style-type: none"> <li>- Enter the defect type from the defect type list summarized in the top left corner of the form.</li> <li>- Use your best judgment in selecting which type applies.</li> </ul>
<b>Inject</b>	<ul style="list-style-type: none"> <li>- Enter the phase when this defect was injected.</li> <li>- Use your best judgment.</li> </ul>
<b>Remove</b>	Enter the phase during which you fixed the defect. (This will generally be the phase when you found the defect.)
<b>Fix Time</b>	<ul style="list-style-type: none"> <li>- Enter the time that you took to find and fix the defect.</li> <li>- This time can be determined by stopwatch or by judgment.</li> </ul>
<b>Fix Ref.</b>	<ul style="list-style-type: none"> <li>- If you or someone else injected this defect while fixing another defect, record the number of the improperly fixed defect.</li> <li>- If you cannot identify the defect number, enter an X.</li> </ul>
<b>Description</b>	Write a succinct description of the defect that is clear enough to later remind you about the error and help you to remember why you made it.

**PSP Defect Type Standard**

Type Number	Type Name	Description
10	Documentation	Comments, messages
20	Syntax	Spelling, punctuation, typos, instruction formats
30	Build, Package	Change management, library, version control
40	Assignment	Declaration, duplicate names, scope, limits
50	Interface	Procedure calls and references, I/O, user formats
60	Checking	Error messages, inadequate checks
70	Data	Structure, content
80	Function	Logic, pointers, loops, recursion, computation, function defects
90	System	Configuration, timing, memory
100	Environment	Design, compile, test, or other support system problems

**PSP0.1 Process Script**

<b>Purpose</b>	To guide the development of module-level programs	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Problem description</li> <li>- PSP0.1 Project Plan Summary form</li> <li>- Time and Defect Recording logs</li> <li>- Defect Type, <b>Coding, and Size Measurement</b> standards</li> <li>- Stopwatch (optional)</li> </ul>	
<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Planning	<ul style="list-style-type: none"> <li>- Produce or obtain a requirements statement.</li> <li>- <b>Estimate the added and modified size of this program.</b></li> <li>- Estimate the required development time.</li> <li>- Enter the plan data in the Project Plan Summary form.</li> <li>- Complete the Time Recording log.</li> </ul>
2	Development	<ul style="list-style-type: none"> <li>- Design the program.</li> <li>- Implement the design.</li> <li>- Compile the program, and fix and log all defects found.</li> <li>- Test the program, and fix and log all defects found.</li> <li>- Complete the Time Recording log.</li> </ul>
3	Postmortem	Complete the Project Plan Summary form with actual time, defect, and size data.
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- A thoroughly tested program</li> <li>- Completed Project Plan Summary form with estimated and actual data</li> <li>- <b>Completed PIP forms</b></li> <li>- Completed Time and Defect Recording logs</li> </ul>	

**PSP0.1 Planning Script**

<b>Purpose</b>	To guide the PSP planning process
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Problem description</li> <li>- PSP0.1 Project Plan Summary form</li> <li>- Time Recording log</li> </ul>

<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Program Requirements	<ul style="list-style-type: none"> <li>- Produce or obtain a requirements statement for the program.</li> <li>- Ensure that the requirements statement is clear and unambiguous.</li> <li>- Resolve any questions.</li> </ul>
2	<b>Size Estimate</b>	<b><i>Make your best estimate of the added and modified size of this program.</i></b>
3	Resource Estimate	<ul style="list-style-type: none"> <li>- Make your best estimate of the time required to develop this program.</li> <li>- <b><i>Using the To Date % from the most recently developed program as a guide, distribute the development time over the planned project phases.</i></b></li> </ul>

<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- Documented requirements statement</li> <li>- Completed Project Plan Summary form with estimated <b><i>program size and</i></b> development time data</li> <li>- Completed Time Recording log</li> </ul>
----------------------	--

**PSP0.1 Development Script**

<b>Purpose</b>	To guide the development of small programs	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Requirements statement</li> <li>- Project Plan Summary form with estimated program <b>size and</b> development time</li> <li>- Time and Defect Recording logs</li> <li>- Defect Type standard <b>and Coding standard</b></li> </ul>	
<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Design	<ul style="list-style-type: none"> <li>- Review the requirements and produce a design to meet them.</li> <li>- Record in the Defect Recording log any requirements defects found.</li> <li>- Record time in the Time Recording log.</li> </ul>
2	Code	<ul style="list-style-type: none"> <li>- Implement the design <b>following the Coding standard.</b></li> <li>- Record in the Defect Recording log any requirements or design defects found.</li> <li>- Record time in the Time Recording log.</li> </ul>
3	Compile	<ul style="list-style-type: none"> <li>- Compile the program until there are no compile errors.</li> <li>- Fix all defects found.</li> <li>- Record defects in the Defect Recording log.</li> <li>- Record time in the Time Recording log.</li> </ul>
4	Test	<ul style="list-style-type: none"> <li>- Test until all tests run without error.</li> <li>- Fix all defects found.</li> <li>- Record defects in the Defect Recording log.</li> <li>- Record time in the Time Recording log.</li> </ul>
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- A thoroughly tested program <b>that conforms to the Coding standard</b></li> <li>- Completed Time and Defect Recording logs</li> </ul>	

**PSP0.1 Postmortem Script**

<b>Purpose</b>	To guide the PSP postmortem process	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Problem description and requirements statement</li> <li>- Project Plan Summary form with program size and development time data</li> <li>- Completed Time and Defect Recording logs</li> <li>- A tested and running program <b><i>that conforms to the coding and size measurement standards</i></b></li> </ul>	
<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Defect Recording	<ul style="list-style-type: none"> <li>- Review the Project Plan Summary to verify that all of the defects found in each phase were recorded.</li> <li>- Using your best recollection, record any omitted defects.</li> </ul>
2	Defect Data Consistency	<ul style="list-style-type: none"> <li>- Check that the data on every defect in the Defect Recording log are accurate and complete.</li> <li>- Verify that the numbers of defects injected and removed per phase are reasonable and correct.</li> <li>- Using your best recollection, correct any missing or incorrect defect data.</li> </ul>
3	<b>Size</b>	<ul style="list-style-type: none"> <li>- <b><i>Count the size of the completed program.</i></b></li> <li>- <b><i>Determine the size of the base, reused, deleted, modified, added, total, added and modified, and new reusable code.</i></b></li> <li>- <b><i>Enter these data in the Project Plan Summary form.</i></b></li> </ul>
4	Time	<ul style="list-style-type: none"> <li>- Review the completed Time Recording log for errors or omissions.</li> <li>- Using your best recollection, correct any missing or incomplete time data.</li> </ul>
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- A thoroughly tested program <b><i>that conforms to the coding and size measurement standards</i></b></li> <li>- Completed Project Plan Summary form</li> <li>- <b><i>Completed PIP forms describing process problems, improvement suggestions, and lessons learned</i></b></li> <li>- Completed Time and Defect Recording logs</li> </ul>	

## PSP0.1 Project Plan Summary

Student			Date	
Program			Program #	
Instructor			Language	
<b>Program Size</b>	<b>Plan</b>	<b>Actual</b>	<b>To Date</b>	
<b>Base (B)</b>				(Measured)
<b>Deleted (D)</b>				(Counted)
<b>Modified (M)</b>				(Counted)
<b>Added (A)</b>				( $T - B + D - R$ )
<b>Reused (R)</b>				(Counted)
<b>Added and Modified (A+M)</b>				( $A + M$ )
<b>Total Size (T)</b>				(Measured)
<b>Total New Reusable</b>				
<b>Time in Phase (min.)</b>	<b>Plan</b>	<b>Actual</b>	<b>To Date</b>	<b>To Date %</b>
Planning				
Design				
Code				
Compile				
Test				
Postmortem				
Total				
<b>Defects Injected</b>	<b>Actual</b>	<b>To Date</b>	<b>To Date %</b>	
Planning				
Design				
Code				
Compile				
Test				
Total Development				
<b>Defects Removed</b>	<b>Actual</b>	<b>To Date</b>	<b>To Date %</b>	
Planning				
Design				
Code				
Compile				
Test				
Total Development				
After Development				

### PSP0.1 Plan Summary Instructions

<b>Purpose</b>	To hold the plan and actual data for programs or program parts
<b>General</b>	<ul style="list-style-type: none"> <li>- <b>Use the most appropriate size measure, either LOC or element count.</b></li> <li>- “To Date” is the total actual to-date values for all products developed.</li> </ul>
<b>Header</b>	<ul style="list-style-type: none"> <li>- Enter your name and the date.</li> <li>- Enter the program name and number.</li> <li>- Enter the instructor’s name and the programming language you are using.</li> </ul>
<b>Program Size</b>	<ul style="list-style-type: none"> <li>- <b>Enter the plan added and modified size value (A+M).</b></li> <li>- <b>Enter actual base, deleted, modified, reused, total, and new reusable size.</b></li> <li>- <b>Calculate actual added size as T-B+D-R and actual added and modified size as A+M.</b></li> <li>- <b>Enter to-date reused, added and modified, total, and new reusable size.</b></li> </ul>
<b>Time in Phase</b>	<ul style="list-style-type: none"> <li>- Enter the estimated total development time.</li> <li>- <b>Distribute the estimated total time across the development phases according to the To Date % for the most recently developed program.</b></li> <li>- Enter the actual time by phase and the total time.</li> <li>- To Date: Enter the sum of the actual times for this program plus the to-date times from the most recently developed program.</li> <li>- To Date %: Enter the percentage of to-date time in each phase.</li> </ul>
<b>Defects Injected</b>	<ul style="list-style-type: none"> <li>- Enter the actual defects by phase and the total actual defects.</li> <li>- To Date: Enter the sum of the actual defects injected by phase and the to-date values for the most recent previously developed program.</li> <li>- To Date %: Enter the percentage of the to-date defects injected by phase.</li> </ul>
<b>Defects Removed</b>	<ul style="list-style-type: none"> <li>- To Date: enter the actual defects removed by phase plus the to-date values for the most recent previously developed program.</li> <li>- To Date %: Enter the percentage of the to-date defects removed by phase.</li> <li>- After development, record any defects subsequently found during program testing, use, reuse, or modification.</li> </ul>

**PSP Process Improvement Proposal (PIP)**

Student \_\_\_\_\_ Date \_\_\_\_\_  
Program \_\_\_\_\_ Program # \_\_\_\_\_  
Instructor \_\_\_\_\_ Language \_\_\_\_\_

**Problem Description**

Briefly describe the problems that you encountered.

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

**Proposal Description**

Briefly describe the process improvements that you propose.

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

**Other Notes and Comments**

Note any other comments or observations that describe your experiences or improvement ideas.

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

**PSP Process Improvement Proposal (PIP) Instructions**

<b>Purpose</b>	<ul style="list-style-type: none"> <li>- To provide a way to record process problems and improvement ideas</li> <li>- To provide an orderly record of your process improvement ideas</li> <li>- To record any other noteworthy observations</li> </ul>
<b>General</b>	<p>Use the PIP form to</p> <ul style="list-style-type: none"> <li>- record process improvement ideas as they occur to you</li> <li>- establish priorities for your improvement plans</li> <li>- describe lessons learned and unusual conditions</li> </ul> <p>Keep PIP forms on hand while using the PSP.</p> <ul style="list-style-type: none"> <li>- Record process problems even without proposed solutions.</li> <li>- Submit a PIP with each PSP assignment report.</li> </ul>
<b>Header</b>	<ul style="list-style-type: none"> <li>- Enter your name and the date.</li> <li>- Enter the program name and number.</li> <li>- Enter the instructor's name and the programming language you are using.</li> </ul>
<b>Problem Description</b>	Briefly describe any problems or experiences that led to this PIP.
<b>Proposal Description</b>	Describe the proposed improvement as explicitly as possible.
<b>Other Notes and Comments</b>	<p>Briefly describe any other observations or facts that would later help you to</p> <ul style="list-style-type: none"> <li>- remember what you did while writing this program</li> <li>- remember an idea for a future improvement</li> <li>- explain to your instructor something you did and why you did it</li> </ul>

## C++ Coding Standard

<b>Purpose</b>	To guide implementation of C++ programs
<b>Program Headers</b>	Begin all programs with a descriptive header.
<b>Header Format</b>	<pre>***** /* Program Assignment: the program number */ /* Name:           your name */ /* Date:           the date you started developing the program */ /* Description:    a short description of the program and what it does */ *****</pre>
<b>Listing Contents</b>	Provide a summary of the listing contents
<b>Contents Example</b>	<pre>***** /* Listing Contents:  *  Reuse instructions  *  Modification instructions  *  Compilation instructions  *  Includes  *  Class declarations:  *  CData  *  ASet  *  Source code in c:/classes/CData.cpp:  *  CData  *  CData()  *  Empty() *****</pre>
<b>Reuse Instructions</b>	<ul style="list-style-type: none"> <li>- Describe how the program is used: declaration format, parameter values, types, and formats.</li> <li>- Provide warnings of illegal values, overflow conditions, or other conditions that could potentially result in improper operation.</li> </ul>
<b>Reuse Instruction Example</b>	<pre>***** /* Reuse instructions  *  int PrintLine(char *line_of_character)  *  Purpose: to print string, 'line_of_character', on one print line  *  Limitations: the line length must not exceed LINE_LENGTH  *  Return 0 if printer not ready to print, else 1 *****</pre>
<b>Identifiers</b>	Use descriptive names for all variable, function names, constants, and other identifiers. Avoid abbreviations or single-letter variables.
<b>Identifier Example</b>	Int number_of_students; /* This is GOOD */ Float: x4, j, ftave; /* This is BAD */
<b>Comments</b>	<ul style="list-style-type: none"> <li>- Document the code so the reader can understand its operation.</li> <li>- Comments should explain both the purpose and behavior of the code.</li> <li>- Comment variable declarations to indicate their purpose.</li> </ul>
<b>Good Comment</b>	If(record_count > limit) /* have all records been processed? */
<b>Bad Comment</b>	If(record_count > limit) /* check if record count exceeds limit */
<b>Major Sections</b>	Precede major program sections by a block comment that describes the processing done in the next section.
<b>Example</b>	<pre>***** /* The program section examines the contents of the array 'grades' and calcu-  * lates the average class grade. *****</pre>
<b>Blank Spaces</b>	<ul style="list-style-type: none"> <li>- Write programs with sufficient spacing so they do not appear crowded.</li> <li>- Separate every program construct with at least one space.</li> </ul>

(continued)

**C++ Coding Standard (continued)**

<b>Indenting</b>	- Indent each brace level from the preceding level. - Open and close braces should be on lines by themselves and aligned.
<b>Indenting Example</b>	while (miss_distance > threshold) { success_code = move_robot (target _location); if (success_code == MOVE_FAILED) { printf("The robot move has failed.\n"); } }
<b>Capitalization</b>	- Capitalize all defines. - Lowercase all other identifiers and reserved words. - To make them readable, user messages may use mixed case.
<b>Capitalization Examples</b>	#define DEFAULT-NUMBER-OF-STUDENTS 15 int class-size = DEFAULT-NUMBER-OF-STUDENTS;

**PSP1 Process Script**

<b>Purpose</b>	To guide the development of module-level programs	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Problem description</li> <li>- PSP1 Project Plan Summary form</li> <li>- <b><i>Size Estimating template</i></b></li> <li>- <b><i>Historical size and time data (estimated and actual)</i></b></li> <li>- Time and Defect Recording logs</li> <li>- Defect Type, Coding, and Size Measurement standards</li> <li>- Stopwatch (optional)</li> </ul>	
<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Planning	<ul style="list-style-type: none"> <li>- Produce or obtain a requirements statement.</li> <li>- <b><i>Use the PROBE method</i></b> to estimate the added and modified size of this program.</li> <li>- <b><i>Complete the Size Estimating template.</i></b></li> <li>- <b><i>Use the PROBE method</i></b> to estimate the required development time.</li> <li>- Enter the plan data in the Project Plan Summary form.</li> <li>- Complete the Time Recording log.</li> </ul>
2	Development	<ul style="list-style-type: none"> <li>- Design the program.</li> <li>- Implement the design.</li> <li>- Compile the program, and fix and log all defects found.</li> <li>- Test the program, and fix and log all defects found.</li> <li>- Complete the Time Recording log.</li> </ul>
3	Postmortem	Complete the Project Plan Summary form with actual time, defect, and size data.
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- A thoroughly tested program</li> <li>- Completed Project Plan Summary form with estimated and actual data</li> <li>- <b><i>Completed Size Estimating template</i></b></li> <li>- <b><i>Completed Test Report template</i></b></li> <li>- Completed PIP forms</li> <li>- Completed Time and Defect Recording logs</li> </ul>	

**PSP1 Planning Script**

<b>Purpose</b>	To guide the PSP planning process	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Problem description</li> <li>- PSP1 Project Plan Summary form</li> <li>- <b><i>Size Estimating template</i></b></li> <li>- <b><i>Historical size and time data (estimated and actual)</i></b></li> <li>- Time Recording log</li> </ul>	
<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Program Requirements	<ul style="list-style-type: none"> <li>- Produce or obtain a requirements statement for the program.</li> <li>- Ensure that the requirements statement is clear and unambiguous.</li> <li>- Resolve any questions.</li> </ul>
2	Size Estimate	<ul style="list-style-type: none"> <li>- <b><i>Produce a program conceptual design.</i></b></li> <li>- <b><i>Use the PROBE method to estimate the added and modified size of this program.</i></b></li> <li>- <b><i>Complete the Size Estimating template and Project Plan Summary form.</i></b></li> </ul>
3	Resource Estimate	<ul style="list-style-type: none"> <li>- <b><i>Use the PROBE method to estimate the time required to develop this program.</i></b></li> <li>- Using the <i>To Date %</i> from the most recently developed program as a guide, distribute the development time over the planned project phases.</li> </ul>
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- Documented requirements statement</li> <li>- <b><i>Program conceptual design</i></b></li> <li>- <b><i>Completed Size Estimating template</i></b></li> <li>- Completed Project Plan Summary form with estimated program size and development time data</li> <li>- Completed Time Recording log</li> </ul>	

**PSP1 Development Script**

<b>Purpose</b>	To guide the development of small programs	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Requirements statement</li> <li>- Project Plan Summary form with estimated program size and development time</li> <li>- Time and Defect Recording logs</li> <li>- Defect Type standard and Coding standard</li> </ul>	
<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Design	<ul style="list-style-type: none"> <li>- Review the requirements and produce a design to meet them.</li> <li>- Record in the Defect Recording log any requirements defects found.</li> <li>- Record time in the Time Recording log.</li> </ul>
2	Code	<ul style="list-style-type: none"> <li>- Implement the design following the Coding standard.</li> <li>- Record in the Defect Recording log any requirements or design defects found.</li> <li>- Record time in the Time Recording log.</li> </ul>
3	Compile	<ul style="list-style-type: none"> <li>- Compile the program until there are no compile errors.</li> <li>- Fix all defects found.</li> <li>- Record defects in the Defect Recording log.</li> <li>- Record time in the Time Recording log.</li> </ul>
4	Test	<ul style="list-style-type: none"> <li>- Test until all tests run without error.</li> <li>- Fix all defects found.</li> <li>- Record defects in the Defect Recording log.</li> <li>- Record time in the Time Recording log.</li> <li>- <b><i>Complete a Test Report template on the tests conducted and the results obtained.</i></b></li> </ul>
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- A thoroughly tested program that conforms to the Coding standard</li> <li>- <b><i>Completed Test Report template</i></b></li> <li>- Completed Time and Defect Recording logs</li> </ul>	

**PSP1 Postmortem Script**

<b>Purpose</b>	To guide the PSP postmortem process	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Problem description and requirements statement</li> <li>- Project Plan Summary form with program size and development time data</li> <li>- <b><i>Completed Test Report template</i></b></li> <li>- Completed Time and Defect Recording logs</li> <li>- A tested and running program that conforms to the coding and size measurement standards</li> </ul>	
<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Defect Recording	<ul style="list-style-type: none"> <li>- Review the Project Plan Summary to verify that all of the defects found in each phase were recorded.</li> <li>- Using your best recollection, record any omitted defects.</li> </ul>
2	Defect Data Consistency	<ul style="list-style-type: none"> <li>- Check that the data on every defect in the Defect Recording log are accurate and complete.</li> <li>- Verify that the numbers of defects injected and removed per phase are reasonable and correct.</li> <li>- Using your best recollection, correct any missing or incorrect defect data.</li> </ul>
3	Size	<ul style="list-style-type: none"> <li>- Count the size of the completed program.</li> <li>- Determine the size of the base, reused, deleted, modified, added, total, added and modified, and new reusable code.</li> <li>- Enter these data in the Project Plan Summary form.</li> </ul>
4	Time	<ul style="list-style-type: none"> <li>- Review the completed Time Recording log for errors or omissions.</li> <li>- Using your best recollection, correct any missing or incomplete time data.</li> </ul>
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- A thoroughly tested program that conforms to the coding and size measurement standards</li> <li>- <b><i>Completed Test Report template</i></b></li> <li>- Completed Project Plan Summary form</li> <li>- Completed PIP forms describing process problems, improvement suggestions, and lessons learned</li> <li>- Completed Time and Defect Recording logs</li> </ul>	

**Example PSP1 Project Plan Summary**

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

<b>Summary Size/Hour</b>	<b>Plan</b>	<b>Actual</b>	<b>To Date</b>
<b>Program Size</b>	<b>Plan</b>	<b>Actual</b>	<b>To Date</b>
Base (B)	(Measured)	(Measured)	
Deleted (D)	(Estimated)	(Counted)	
Modified (M)	(Estimated)	(Counted)	
Added (A)	(A+M - M)	(T - B + D - R)	
Reused (R)	(Estimated)	(Counted)	
Added and Modified (A+M)	(Projected)	(A + M)	
Total Size (T)	(A+M + B - M - D + R)	(Measured)	
Total New Reusable			
<b>Estimated Proxy Size (E)</b>			
<b>Time in Phase (min.)</b>	<b>Plan</b>	<b>Actual</b>	<b>To Date</b>
Planning			
Design			
Code			
Compile			
Test			
Postmortem			
Total			
<b>Defects Injected</b>	<b>Actual</b>	<b>To Date</b>	<b>To Date %</b>
Planning			
Design			
Code			
Compile			
Test			
Total Development			
<b>Defects Removed</b>	<b>Actual</b>	<b>To Date</b>	<b>To Date %</b>
Planning			
Design			
Code			
Compile			
Test			
Total Development			
After Development			

### PSP1 Plan Summary Instructions

<b>Purpose</b>	To hold the plan and actual data for programs or program parts
<b>General</b>	<ul style="list-style-type: none"> <li>- Use the most appropriate size measure, either LOC or element count.</li> <li>- “To Date” is the total actual to-date values for all products developed.</li> <li>- <b>A part could be a module, component, product, or system.</b></li> </ul>
<b>Header</b>	<ul style="list-style-type: none"> <li>- Enter your name and the date.</li> <li>- Enter the program name and number.</li> <li>- Enter the instructor’s name and the programming language you are using.</li> </ul>
<b>Summary</b>	<ul style="list-style-type: none"> <li>- <b>Enter the added and modified size per hour planned, actual, and to-date.</b></li> </ul>
<b>Program Size</b>	<ul style="list-style-type: none"> <li>- <b>Enter plan base, deleted, modified, reused, new reusable, and total size from the Size Estimating template.</b></li> <li>- Enter the plan added and modified size value (A+M) <b>from projected added and modified size (P) on the Size Estimating template.</b></li> <li>- <b>from the Size Estimating template.</b></li> <li>- <b>Calculate plan added size as A+M – M.</b></li> <li>- <b>Enter estimated proxy size (E) from the Size Estimating template.</b></li> <li>- Enter actual base, deleted, modified, reused, total, and new reusable size Calculate actual added size as T-B+D-R and actual added and modified size as A+M.</li> <li>- Enter to-date reused, added and modified, total, and new reusable size.</li> </ul>
<b>Time in Phase</b>	<ul style="list-style-type: none"> <li>- Enter <b>plan total time in phase from</b> the estimated total development time <b>on the Size Estimating template.</b></li> <li>- Distribute the estimated total time across the development phases according to the To Date % for the most recently developed program.</li> <li>- Enter the actual time by phase and the total time.</li> <li>- To Date: Enter the sum of the actual times for this program plus the to-date times from the most recently developed program.</li> <li>- To Date %: Enter the percentage of to-date time in each phase.</li> </ul>
<b>Defects Injected</b>	<ul style="list-style-type: none"> <li>- Enter the actual defects by phase and the total actual defects.</li> <li>- To Date: Enter the sum of the actual defects injected by phase and the to-date values for the most recent previously developed program.</li> <li>- To Date %: Enter the percentage of the to-date defects injected by phase.</li> </ul>
<b>Defects Removed</b>	<ul style="list-style-type: none"> <li>- To Date: Enter the actual defects removed by phase plus the to-date values for the most recent previously developed program.</li> <li>- To Date %: Enter the percentage of the to-date defects removed by phase.</li> <li>- After development, record any defects subsequently found during program testing, use, reuse, or modification.</li> </ul>

### PROBE Estimating Script

<b>Purpose</b>	To guide the size and time estimating process using the PROBE method
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Requirements statement</li> <li>- Size Estimating template and instructions</li> <li>- Size per item data for part types</li> <li>- Time Recording log</li> <li>- Historical size and time data</li> </ul>
<b>General</b>	<ul style="list-style-type: none"> <li>- This script assumes that you are using added and modified size data as the size-accounting types for making size and time estimates.</li> <li>- If you choose some other size-accounting types, replace every “added and modified” in this script with the size-accounting types of your choice.</li> </ul>

<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Conceptual Design	Review the requirements and produce a conceptual design.
2	Parts Additions	Follow the Size Estimating Template instructions to estimate the parts additions and the new reusable parts sizes.
3	Base Parts and Reused Parts	<ul style="list-style-type: none"> <li>- For the base program, estimate the size of the base, deleted, modified, and added code.</li> <li>- Measure and/or estimate the size of the parts to be reused.</li> </ul>
4	Size Estimating Procedure	<ul style="list-style-type: none"> <li>- If you have sufficient estimated proxy size and actual added and modified size data (three or more points that correlate), use procedure 4A.</li> <li>- If you do not have sufficient estimated data but have sufficient plan added and modified and actual added and modified size data (three or more points that correlate), use procedure 4B.</li> <li>- If you have insufficient data or they do not correlate, use procedure 4C.</li> <li>- If you have no historical data, use procedure 4D.</li> </ul>
4A	Size Estimating Procedure 4A	<ul style="list-style-type: none"> <li>- Using the linear-regression method, calculate the <math>\beta_0</math> and <math>\beta_1</math> parameters from the estimated proxy size and actual added and modified size data.</li> <li>- If the absolute value of <math>\beta_0</math> is not near 0 (less than about 25% of the expected size of the new program), or <math>\beta_1</math> is not near 1.0 (between about 0.5 and 2.0), use procedure 4B.</li> </ul>
4B	Size Estimating Procedure 4B	<ul style="list-style-type: none"> <li>- Using the linear-regression method, calculate the <math>\beta_0</math> and <math>\beta_1</math> parameters from the plan added and modified size and actual added and modified size data.</li> <li>- If the absolute value of <math>\beta_0</math> is not near 0 (less than about 25% of the expected size of the new program), or <math>\beta_1</math> is not near 1.0 (between about 0.5 and 2.0), use procedure 4C.</li> </ul>
4C	Size Estimating Procedure 4C	If you have any data on plan added and modified size and actual added and modified size, set $\beta_0 = 0$ and $\beta_1 = (\text{actual total added and modified size to date}/\text{plan total added and modified size to date})$ .
4D	Size Estimating Procedure 4D	If you have no historical data, use your judgment to estimate added and modified size.

**(continued)**

**PROBE Script (continued)**

<b>Step</b>	<b>Activities</b>	<b>Description</b>
5	Time Estimating Procedure	<ul style="list-style-type: none"> <li>- If you have sufficient estimated proxy size and actual development time data (three or more points that correlate), use procedure 5A.</li> <li>- If you do not have sufficient estimated size data but have sufficient plan added and modified size and actual development time data (three or more points that correlate), use procedure 5B.</li> <li>- If you have insufficient data or they do not correlate, use procedure 5C.</li> <li>- If you have no historical data, use procedure 5D.</li> </ul>
5A	Time Estimating Procedure 5A	<ul style="list-style-type: none"> <li>- Using the linear-regression method, calculate the <math>\beta_0</math> and <math>\beta_1</math> parameters from the estimated proxy size and actual total development time data.</li> <li>- If <math>\beta_0</math> is not near 0 (substantially smaller than the expected development time for the new program), or <math>\beta_1</math> is not within 50% of 1/(historical productivity), use procedure 5B.</li> </ul>
5B	Time Estimating Procedure 5B	<ul style="list-style-type: none"> <li>- Using the linear-regression method, calculate the <math>\beta_0</math> and <math>\beta_1</math> regression parameters from the plan added and modified size and actual total development time data.</li> <li>- If <math>\beta_0</math> is not near 0 (substantially smaller than the expected development time for the new program), or <math>\beta_1</math> is not within 50% of 1/(historical productivity), use procedure 5C.</li> </ul>
5C	Time Estimating Procedure 5C	<ul style="list-style-type: none"> <li>- If you have data on estimated – added and modified size and actual development time, set <math>\beta_0 = 0</math> and <math>\beta_1 = (\text{actual total development time to date}/\text{estimated – total added and modified size to date})</math>.</li> <li>- If you have data on plan – added and modified size and actual development time, set <math>\beta_0 = 0</math> and <math>\beta_1 = (\text{actual total development time to date}/\text{plan total added and modified size to date})</math>.</li> <li>- If you only have actual time and size data, set <math>\beta_0 = 0</math> and <math>\beta_1 = (\text{actual total development time to date}/\text{actual total added and modified size to date})</math>.</li> </ul>
5D	Time Estimating Procedure 5D	If you have no historical data, use your judgment to estimate the development time from the estimated added and modified size.
6	Time and Size Prediction Intervals	<ul style="list-style-type: none"> <li>- If you used regression method A or B, calculate the 70% prediction intervals for the time and size estimates.</li> <li>- If you did not use the regression method or do not know how to calculate the prediction interval, calculate the minimum and maximum development time estimate limits from your historical maximum and minimum productivity for the programs written to date.</li> </ul>
<b>Exit Criteria</b>		<ul style="list-style-type: none"> <li>- Completed estimated and actual entries for all pertinent size categories</li> <li>- Completed PROBE Calculation Worksheet with size and time entries</li> <li>- Plan and actual values entered on the Project Plan Summary</li> </ul>

**Test Report Template**

Student	<hr/>	Date	<hr/>
Program	<hr/>	Program #	<hr/>
Instructor	<hr/>	Language	<hr/>

Test Name/Number	<hr/>
Test Objective	<hr/>
Test Description	<hr/> <hr/> <hr/> <hr/>
Test Conditions	<hr/> <hr/> <hr/> <hr/>
Expected Results	<hr/> <hr/> <hr/> <hr/>
Actual Results	<hr/> <hr/> <hr/> <hr/>
Test Name/Number	<hr/>
Test Objective	<hr/>
Test Description	<hr/> <hr/> <hr/> <hr/>
Test Conditions	<hr/> <hr/> <hr/> <hr/>
Expected Results	<hr/> <hr/> <hr/> <hr/>
Actual Results	<hr/> <hr/> <hr/> <hr/>

### Test Report Template Instructions

<b>Purpose</b>	<ul style="list-style-type: none"> <li>- To maintain a record of the tests run and the results obtained</li> <li>- To be sufficiently complete so that you can later re-run the same tests and get the same results</li> <li>- To facilitate regression testing of modified or reused programs</li> </ul>
<b>General</b>	<ul style="list-style-type: none"> <li>- Expand this table or use multiple copies as needed.</li> <li>- Report all the tests that were successfully run.</li> <li>- Be as brief and concise as possible.</li> </ul>
<b>Header</b>	<ul style="list-style-type: none"> <li>- Enter your name and the date.</li> <li>- Enter the program name and number.</li> <li>- Enter the instructor's name and the programming language you are using.</li> </ul>
<b>Test Name/Number</b>	<p>Uniquely identify each test for each program.</p> <ul style="list-style-type: none"> <li>- the same tests with different data</li> <li>- the same data with different tests</li> </ul>
<b>Test Objective</b>	Briefly describe the objective of the test.
<b>Test Description</b>	Describe each test's data and procedures in sufficient detail to facilitate its later use as a regression test.
<b>Test Conditions</b>	<ul style="list-style-type: none"> <li>- List any special configuration, timing, fix, or other conditions of the test.</li> <li>- When multiple tests are run with different parameters or under varying conditions, separately list each.</li> </ul>
<b>Expected Results</b>	List the results that the test should produce if it runs properly.
<b>Actual Results</b>	List the results that were actually produced.

**Size Estimating Template**

Student \_\_\_\_\_ Date \_\_\_\_\_  
 Program \_\_\_\_\_ Program # \_\_\_\_\_  
 Instructor \_\_\_\_\_ Language \_\_\_\_\_  
 Size Measure \_\_\_\_\_

Base Parts	Estimated			
	Base	Deleted	Modified	Added
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
<b>Total</b>	B _____	D _____	M _____	BA _____

Base Parts	Actual			
	Base	Deleted	Modified	Added
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
<b>Total</b>	_____	_____	_____	_____

Parts Additions	Type	Estimated		Actual	
		Items	Rel. Size	Size*	Size*
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
<b>Total</b>	_____	_____	_____	PA _____	_____

Reused Parts	Estimated Size		Actual Size
		Size	
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
<b>Total</b>	R _____	_____	_____

(continued)

**Size Estimating Template (continued)**

Student	Program	Size	Time
<b>PROBE Calculation Worksheet (Added and Modified)</b>			
Added size (A):	$A = BA + PA$	_____	_____
Estimated Proxy Size (E):	$E = BA + PA + M$	_____	_____
PROBE estimating basis used: (A, B, C, or D)	_____	_____	_____
Correlation: ( $R^2$ )	_____	_____	_____
Regression Parameters:	$\beta_0$ Size and Time	_____	_____
Regression Parameters:	$\beta_1$ Size and Time	_____	_____
Projected Added and Modified Size (P):	$P = \beta_{0_{size}} + \beta_{1_{size}} * E$	_____	_____
Estimated Total Size (T):	$T = P + B - D - M + R$	_____	_____
Estimated Total New Reusable (NR):	sum of * items	_____	_____
Estimated Total Development Time:	$Time = \beta_{0_{time}} + \beta_{1_{time}} * E$	_____	_____
Prediction Range:	Range	_____	_____
Upper Prediction Interval:	$UPI = P + Range$	_____	_____
Lower Prediction Interval:	$LPI = P - Range$	_____	_____
Prediction Interval Percent:	_____	_____	_____

### Size Estimating Template Instructions

<b>Purpose</b>	Use this form with the PROBE method to make size estimates.
<b>General</b>	<ul style="list-style-type: none"> <li>- A part could be a module, component, product, or system.</li> <li>- Where parts have a substructure of methods, procedures, functions, or similar elements, these lowest-level elements are called items.</li> <li>- Size values are assumed to be in the unit specified in size measure.</li> <li>- Avoid confusing base size with reuse size.</li> <li>- Reuse parts must be used without modification.</li> <li>- Use base size if additions, modifications, or deletions are planned.</li> <li>- If a part is estimated but not produced, enter its actual values as zero.</li> <li>- If a part is produced that was not estimated, enter it using zero for its planned values.</li> </ul>
<b>Header</b>	<ul style="list-style-type: none"> <li>- Enter your name and the date.</li> <li>- Enter the program name and number.</li> <li>- Enter the instructor's name and the programming language you are using.</li> <li>- Enter the size measure you are using.</li> </ul>
<b>Base Parts</b>	<p>If this is a modification or enhancement of an existing product</p> <ul style="list-style-type: none"> <li>- measure and enter the base size (more than one product may be entered as base)</li> <li>- estimate and enter the size of the deleted, modified, and added size to the base program</li> </ul> <p>After development, measure and enter the actual size of the base program and any deletions, modifications, or additions.</p>
<b>Parts Additions</b>	<p>If you plan to add newly developed parts</p> <ul style="list-style-type: none"> <li>- enter the part name, type, number of items (or methods), and relative size</li> <li>- for each part, get the size per item from the appropriate relative size table, multiply this value by the number of items, and enter in estimated size</li> <li>- put an asterisk next to the estimated size of any new-reusable additions</li> </ul> <p>After development, measure and enter</p> <ul style="list-style-type: none"> <li>- the actual size of each new part or new part items</li> <li>- the number of items for each new part</li> </ul>
<b>Reused Parts</b>	<p>If you plan to include reused parts, enter the</p> <ul style="list-style-type: none"> <li>- name of each unmodified reused part</li> <li>- size of each unmodified reused part</li> </ul> <p>After development, enter the actual size of each unmodified reused part.</p>

### PROBE Calculation Worksheet Instructions

<b>Purpose</b>	Use this form with the PROBE method to make size and resource estimate calculations.
<b>General</b>	<p>The PROBE method can be used for many kinds of estimates. Where development time correlates with added and modified size</p> <ul style="list-style-type: none"> <li>- use the Added and Modified Calculation Worksheet</li> <li>- enter the resulting estimates in the Project Plan Summary</li> <li>- enter the projected added and modified value (P) in the added and modified plan space in the Project Plan Summary</li> </ul> <p>If development time correlates with some other combination of size-accounting types</p> <ul style="list-style-type: none"> <li>- define and use a new PROBE Calculation Worksheet</li> <li>- enter the resulting estimates in the Project Plan Summary</li> <li>- use the selected combination of size accounting types to calculate the projected size value (P)</li> <li>- enter this P value in the Project Plan Summary for the appropriate plan size for the size-accounting types being used</li> </ul>
<b>PROBE Calculations: Size (Added and Modified)</b>	<ul style="list-style-type: none"> <li>- Added Size (A): Total the added base code (BA) and Parts Additions (PA) to get Added Size (A).</li> <li>- Estimated Proxy Size (E): Total the added (A) and modified (M) sizes and enter as (E).</li> <li>- PROBE Estimating Basis Used: Analyze the available historical data and select the appropriate PROBE estimating basis (A, B, C, or D).</li> <li>- Correlation: If PROBE estimating basis A or B is selected, enter the correlation value (<math>R^2</math>) for both size and time.</li> <li>- Regression Parameters: Follow the procedure in the PROBE script to calculate the size and time regression parameters (<math>\beta_0</math> and <math>\beta_1</math>), and enter them in the indicated fields.</li> <li>- Projected Added and Modified Size (P): Using the size regression parameters and estimated proxy size (E), calculate the projected added and modified size (P) as <math>P = \beta_{0Size} + \beta_{1Size} * E</math>.</li> <li>- Estimated Total Size (T): Calculate the estimated total size as <math>T = P+B-D-M+R</math>.</li> <li>- Estimated Total New Reusable (NR): Total and enter the new reusable items marked with *.</li> </ul>
<b>PROBE Calculations: Time (Added and Modified)</b>	<ul style="list-style-type: none"> <li>- PROBE Estimating Basis Used: Analyze the available historical data and select the appropriate PROBE estimating basis (A, B, C, or D).</li> <li>- Estimated Total Development Time: Using the time regression parameters and estimated proxy size (E), calculate the estimated development time as <math>Time = \beta_{0Time} + \beta_{1Time} * E</math>.</li> </ul>
<b>PROBE Calculations: Prediction Range</b>	<ul style="list-style-type: none"> <li>- Calculate and enter the prediction range for both the size and time estimates.</li> <li>- Calculate the upper (UPI) and lower (LPI) prediction intervals for both the size and time estimates.</li> <li>- Prediction Interval Percent: List the probability percent used to calculate the prediction intervals (70% or 90%).</li> </ul>
<b>After Development (Added and Modified)</b>	Enter the actual sizes for base (B), deleted (D), modified (M), and added base code (BA), parts additions (PA), and reused parts (R).

### PSP1.1 Process Script

<b>Purpose</b>	To guide the development of module-level programs
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Problem description</li> <li>- PSP1.1 Project Plan Summary form</li> <li>- Size Estimating template</li> <li>- Historical size and time data (estimated and actual)</li> <li>- Time and Defect Recording logs</li> <li>- Defect Type, Coding, and Size Measurement standards</li> <li>- Stopwatch (optional)</li> </ul>

<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Planning	<ul style="list-style-type: none"> <li>- Produce or obtain a requirements statement.</li> <li>- Use the PROBE method to estimate the added and modified size of this program.</li> <li>- Complete the Size Estimating template.</li> <li>- Use the PROBE method to estimate the required development time.</li> <li>- <b><i>Complete a Task Planning template.</i></b></li> <li>- <b><i>Complete a Schedule Planning template.</i></b></li> <li>- Enter the plan data in the Project Plan Summary form.</li> <li>- Complete the Time Recording log.</li> </ul>
2	Development	<ul style="list-style-type: none"> <li>- Design the program.</li> <li>- Implement the design.</li> <li>- Compile the program, and fix and log all defects found.</li> <li>- Test the program, and fix and log all defects found.</li> <li>- <b><i>Complete the Time Recording log.</i></b></li> </ul>
3	Postmortem	Complete the Project Plan Summary form with actual time, defect, and size data.

<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- A thoroughly tested program</li> <li>- Completed Project Plan Summary form with estimated and actual data</li> <li>- Completed Size Estimating <b><i>and Task and Schedule Planning</i></b> templates</li> <li>- Completed Test Report template</li> <li>- Completed PIP forms</li> <li>- Completed Time and Defect Recording logs</li> </ul>
----------------------	--

**PSP1.1 Planning Script**

<b>Purpose</b>	To guide the PSP planning process	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Problem description</li> <li>- PSP1.1 Project Plan Summary form</li> <li>- Size Estimating, <b><i>Task Planning, and Schedule Planning templates</i></b></li> <li>- Historical size and time data (estimated and actual)</li> <li>- Time Recording log</li> </ul>	
<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Program Requirements	<ul style="list-style-type: none"> <li>- Produce or obtain a requirements statement for the program.</li> <li>- Ensure that the requirements statement is clear and unambiguous.</li> <li>- Resolve any questions.</li> </ul>
2	Size Estimate	<ul style="list-style-type: none"> <li>- Produce a program conceptual design.</li> <li>- Use the PROBE method to estimate the added and modified size of this program.</li> <li>- Complete the Size Estimating template and Project Plan Summary form.</li> </ul>
3	Resource Estimate	<ul style="list-style-type: none"> <li>- Use the PROBE method to estimate the time required to develop this program.</li> <li>- Using the <i>To Date %</i> from the most recently developed program as a guide, distribute the development time over the planned project phases.</li> </ul>
4	<b><i>Task and Schedule Planning</i></b>	<b><i>For projects lasting several days or more, complete the Task Planning and Schedule Planning templates.</i></b>
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- Documented requirements statement</li> <li>- Program conceptual design</li> <li>- Completed Size Estimating template</li> <li>- <b><i>For projects lasting several days or more, completed Task and Schedule Planning templates</i></b></li> <li>- Completed Project Plan Summary form with estimated program size and development time data</li> <li>- Completed Time Recording log</li> </ul>	

### PSP1.1 Development Script

<b>Purpose</b>	To guide the development of small programs	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Requirements statement</li> <li>- Project Plan Summary form with estimated program size and development time</li> <li>- <b><i>For projects lasting several days or more, completed Task Planning and Schedule Planning templates</i></b></li> <li>- Time and Defect Recording logs</li> <li>- Defect Type standard and Coding standard</li> </ul>	
<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Design	<ul style="list-style-type: none"> <li>- Review the requirements and produce a design to meet them.</li> <li>- Record in the Defect Recording log any requirements defects found.</li> <li>- Record time in the Time Recording log.</li> </ul>
2	Code	<ul style="list-style-type: none"> <li>- Implement the design following the Coding standard.</li> <li>- Record in the Defect Recording log any requirements or design defects found.</li> <li>- Record time in the Time Recording log.</li> </ul>
3	Compile	<ul style="list-style-type: none"> <li>- Compile the program until there are no compile errors.</li> <li>- Fix all defects found.</li> <li>- Record defects in the Defect Recording log.</li> <li>- Record time in the Time Recording log.</li> </ul>
4	Test	<ul style="list-style-type: none"> <li>- Test until all tests run without error.</li> <li>- Fix all defects found.</li> <li>- Record defects in the Defect Recording log.</li> <li>- Record time in the Time Recording log.</li> <li>- Complete a Test Report template on the tests conducted and the results obtained.</li> </ul>
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- A thoroughly tested program that conforms to the Coding standard</li> <li>- Completed Test Report template</li> <li>- Completed Time and Defect Recording logs</li> </ul>	

**PSP1.1 Postmortem Script**

<b>Purpose</b>	To guide the PSP postmortem process	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Problem description and requirements statement</li> <li>- Project Plan Summary form with program size and development time data</li> <li>- <b><i>For projects lasting several days or more, completed Task Planning and Schedule Planning templates</i></b></li> <li>- Completed Test Report template</li> <li>- Completed Time and Defect Recording logs</li> <li>- A tested and running program that conforms to the coding and size measurement standards</li> </ul>	
<b>Step</b>   <b>Activities</b>   <b>Description</b>		
1	Defect Recording	<ul style="list-style-type: none"> <li>- Review the Project Plan Summary to verify that all of the defects found in each phase were recorded.</li> <li>- Using your best recollection, record any omitted defects.</li> </ul>
2	Defect Data Consistency	<ul style="list-style-type: none"> <li>- Check that the data on every defect in the Defect Recording log are accurate and complete.</li> <li>- Verify that the numbers of defects injected and removed per phase are reasonable and correct.</li> <li>- Using your best recollection, correct any missing or incorrect defect data.</li> </ul>
3	Size	<ul style="list-style-type: none"> <li>- Count the size of the completed program.</li> <li>- Determine the size of the base, reused, deleted, modified, added, total, added and modified, and new reusable code.</li> <li>- Enter these data in the Project Plan Summary form.</li> </ul>
4	Time	<ul style="list-style-type: none"> <li>- Review the completed Time Recording log for errors or omissions.</li> <li>- Using your best recollection, correct any missing or incomplete time data.</li> </ul>
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- A thoroughly tested program that conforms to the coding and size measurement standards</li> <li>- Completed Test Report template</li> <li>- Completed Project Plan Summary form</li> <li>- Completed PIP forms describing process problems, improvement suggestions, and lessons learned</li> <li>- Completed Time and Defect Recording logs</li> </ul>	

**PSP1.1 Project Plan Summary**

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

<b>Summary</b>	<b>Plan</b>	<b>Actual</b>	<b>To Date</b>
Size/Hour	_____	_____	_____
<b>Planned Time</b>	_____	_____	_____
<b>Actual Time</b>	_____	_____	_____
<b>CPI (Cost-Performance Index)</b>	_____	_____	(Planned/Actual)
<b>% Reused</b>	_____	_____	_____
<b>% New Reusable</b>	_____	_____	_____
<b>Program Size</b>	<b>Plan</b>	<b>Actual</b>	<b>To Date</b>
Base (B)	(Measured)	(Measured)	_____
Deleted (D)	(Estimated)	(Counted)	_____
Modified (M)	(Estimated)	(Counted)	_____
Added (A)	(A+M - M)	(T - B + D - R)	_____
Reused (R)	(Estimated)	(Counted)	_____
Added and Modified (A+M)	(Projected)	(A + M)	_____
Total Size (T)	(A+M + B - M - D + R)	(Measured)	_____
Total New Reusable	_____	_____	_____
Estimated Proxy Size (E)	_____	_____	_____

<b>Time in Phase (min.)</b>	<b>Plan</b>	<b>Actual</b>	<b>To Date</b>	<b>To Date %</b>
Planning	_____	_____	_____	_____
Design	_____	_____	_____	_____
Code	_____	_____	_____	_____
Compile	_____	_____	_____	_____
Test	_____	_____	_____	_____
Postmortem	_____	_____	_____	_____
Total	_____	_____	_____	_____

**(continued)**

**PSP1.1 Project Plan Summary (continued)**

Student _____	Program # _____		
<b>Defects Injected</b>	<b>Actual</b>	<b>To Date</b>	<b>To Date %</b>
Planning	_____	_____	_____
Design	_____	_____	_____
Code	_____	_____	_____
Compile	_____	_____	_____
Test	_____	_____	_____
Total Development	_____	_____	_____
<b>Defects Removed</b>	<b>Actual</b>	<b>To Date</b>	<b>To Date %</b>
Planning	_____	_____	_____
Design	_____	_____	_____
Code	_____	_____	_____
Compile	_____	_____	_____
Test	_____	_____	_____
Total Development	_____	_____	_____
After Development	_____	_____	_____

### PSP1.1 Plan Summary Instructions

<b>Purpose</b>	To hold the plan and actual data for programs or program parts
<b>General</b>	<ul style="list-style-type: none"> <li>- Use the most appropriate size measure, either LOC or element count.</li> <li>- “To Date” is the total actual to-date values for all products developed.</li> <li>- A part could be a module, component, product, or system.</li> </ul>
<b>Header</b>	<ul style="list-style-type: none"> <li>- Enter your name and the date.</li> <li>- Enter the program name and number.</li> <li>- Enter the instructor’s name and the programming language you are using.</li> </ul>
<b>Summary</b>	<ul style="list-style-type: none"> <li>- Enter the added and modified size per hour planned, actual, and to-date.</li> <li>- <b><i>Enter the planned and actual times for this program and prior programs.</i></b></li> <li>- <b><i>For planned time to date, use the sum of the current planned time and the planned times for the prior programs.</i></b></li> <li>- <b><i>CPI = (To Date Planned Time)/(To Date Actual Time).</i></b></li> <li>- <b><i>Reused % is reused size as a percentage of total program size.</i></b></li> <li>- <b><i>New Reusable % is new reusable size as a percentage of added and modified size.</i></b></li> </ul>
<b>Program Size</b>	<ul style="list-style-type: none"> <li>- Enter plan base, deleted, modified, reused, new reusable, and total size from the Size Estimating template.</li> <li>- Enter the plan added and modified size value (A+M) from projected added and modified size (P) on the Size Estimating template.</li> <li>- Calculate plan added size as A+M – M.</li> <li>- Enter estimated proxy size (E) from the Size Estimating template.</li> <li>- Enter actual base, deleted, modified, reused, total, and new reusable size from the Size Estimating template.</li> <li>- Calculate actual added size as T-B+D-R and actual added and modified size as A+M.</li> <li>- Enter to-date reused, added and modified, total, and new reusable size.</li> </ul>
<b>Time in Phase</b>	<ul style="list-style-type: none"> <li>- Enter plan total time in phase from the estimated total development time on the Size Estimating template.</li> <li>- Distribute the estimated total time across the development phases according to the To Date % for the most recently developed program.</li> <li>- Enter the actual time by phase and the total time.</li> <li>- To Date: Enter the sum of the actual times for this program plus the to-date times from the most recently developed program.</li> <li>- To Date %: Enter the percentage of to-date time in each phase.</li> </ul>
<b>Defects Injected</b>	<ul style="list-style-type: none"> <li>- Enter the actual defects by phase and the total actual defects.</li> <li>- To Date: Enter the sum of the actual defects injected by phase and the to-date values for the most recent previously developed program.</li> <li>- To Date %: Enter the percentage of the to-date defects injected by phase.</li> </ul>
<b>Defects Removed</b>	<ul style="list-style-type: none"> <li>- To Date: Enter the actual defects removed by phase plus the to-date values for the most recent previously developed program.</li> <li>- To Date %: Enter the percentage of the to-date defects removed by phase.</li> <li>- After development, record any defects subsequently found during program testing, use, reuse, or modification.</li> </ul>

## Task Planning Template

Student \_\_\_\_\_ Date \_\_\_\_\_  
Program \_\_\_\_\_ Program # \_\_\_\_\_  
Instructor \_\_\_\_\_ Language \_\_\_\_\_

### Task Planning Template Instructions

<b>Purpose</b>	<ul style="list-style-type: none"> <li>- To estimate the development time for each project task</li> <li>- To compute the planned value for each project task</li> <li>- To estimate the planned completion date for each task</li> <li>- To provide a basis for tracking schedule progress even when the tasks are not completed in the planned order</li> </ul>
<b>General</b>	<ul style="list-style-type: none"> <li>- Complete the Schedule Planning and Task Planning templates together.</li> <li>- Select tasks that have explicit completion criteria, i.e., plan completed, program compiled and defects corrected, etc.</li> <li>- Expand this template or use multiple pages as needed.</li> <li>- Include every significant task.</li> <li>- Use task names and numbers that support the activity and are consistent with the project work breakdown structure.</li> <li>- Note that most support tools will do the earned-value calculations.</li> </ul>
<b>Header</b>	<ul style="list-style-type: none"> <li>- Enter your name and the date.</li> <li>- Enter the program name and number.</li> <li>- Enter the instructor's name and the programming language you are using.</li> </ul>
<b>Program/Part</b>	Enter the program or part to which the task relates.
<b>Phase</b>	Enter the phase for each task.
<b>Task Name</b>	Enter task names and/or numbers in the order in which you expect to complete them.
<b>Plan: Task Hours</b>	Enter the total planned hours for each task.
<b>Plan: Cumulative Task Hours</b>	Enter the cumulative sum of the total planned task hours.
<b>Plan: Week Due</b>	If the task has a specific due date, enter the week due here.
<b>Plan: Week</b>	<ul style="list-style-type: none"> <li>- On the Schedule template, find the plan cumulative schedule hours entry that equals or just exceeds each cumulative task hours entry on this form.</li> <li>- The week number in that row of the Schedule template is the plan week number for the task on Task template.</li> <li>- If several weeks on the Schedule template have the same cumulative value, enter the earliest week number.</li> </ul>
<b>Plan: Week Predicted</b>	<ul style="list-style-type: none"> <li>- On the Schedule template, find the predicted cumulative earned value entry that equals or just exceeds each cumulative planned value entry on this form.</li> <li>- The week number in that row of the Schedule template is the predicted week number for the task on the Task template.</li> <li>- If several weeks on the Schedule template have the same cumulative value, enter the earliest week number.</li> </ul>
<b>Plan: Planned Value (PV)</b>	<ul style="list-style-type: none"> <li>- Total the planned hours for all tasks.</li> <li>- Find the percentage each task's planned hours is of total hours.</li> <li>- Enter this percentage as the planned value for each task.</li> </ul>
<b>Plan: Cumulative PV</b>	Enter the cumulative sum of the planned values.
<b>Actual: Task Hours</b>	When a task is completed, enter the hours spent on the task.
<b>Actual: Cumulative Earned Value (EV)</b>	<ul style="list-style-type: none"> <li>- Each week, total the EV for all completed tasks and enter that total beside the latest completed task.</li> <li>- Also enter the weekly and cumulative total EV on the Schedule template.</li> </ul>
<b>Actual: Week</b>	As a task is completed, enter the week number it was completed.

# Schedule Planning Template

Student \_\_\_\_\_ Date \_\_\_\_\_  
Program \_\_\_\_\_ Program # \_\_\_\_\_  
Instructor \_\_\_\_\_ Language \_\_\_\_\_

### Schedule Planning Template Instructions

<b>Purpose</b>	- To record the estimated and actual hours expended by calendar period - To relate the task planned value to the calendar schedule
<b>General</b>	- Expand this template or use multiple pages as needed. - Complete in conjunction with the Task Planning template.
<b>Header</b>	- Enter your name and the date. - Enter the program name and number. - Enter the instructor's name and the programming language you are using.
<b>Week No.</b>	- From the project start, enter a week number, typically starting with 1. - For very small projects, it may be more convenient to use days instead of weeks.
<b>Date</b>	- Enter the calendar date for each week. - Pick a standard day in the week (for example, Monday).
<b>Plan: Schedule Hours</b>	- Enter the planned number of schedule hours that you expect to spend working on the project each week. - Consider non-work time such as vacations, holidays, etc. - Consider other committed activities such as e-mail, courses, meetings, and other projects.
<b>Plan: Cumulative Schedule Hours</b>	Enter the planned cumulative schedule hours through each week.
<b>Plan: Cumulative Planned Value</b>	For each week - take the plan cumulative schedule hours from the Schedule template - on the Task template, find the task with nearest equal or lower plan cumulative task hours and note its plan cumulative value - enter this cumulative value in the Schedule template for that week - if the cumulative value for the prior week still applies, enter it again
<b>Actual: Schedule Hours</b>	At the end of each week, enter the actual schedule hours worked in that week.
<b>Actual: Cumulative Schedule Hours</b>	At the end of each week, calculate and enter the actual cumulative schedule hours for the week.
<b>Actual: Week Earned Value</b>	At the end of each week, calculate the total earned value for each task completed during the week and enter here.
<b>Actual: Cumulative Earned Value</b>	At the end of each week, calculate the cumulative earned value for the week.
<b>Predicted: Cumulative Predicted Earned Value</b>	At the end of each week, recalculate the cumulative predicted earned value for the current week through to the end of the schedule. - Enter the current week's actual cumulative earned value as the current week's cumulative predicted earned value. - Calculate the average actual earned value per hour worked on the job to date (Actual Cumulative EV/Actual Cumulative Schedule Hours). - For each week n, starting with the next week, multiply the average earned value per planned hour by the planned hours for week n. Add the result to the cumulative predicted earned value for the preceding week and enter in the cumulative predicted earned value for that week. Repeat for each week until the cumulative predicted earned value reaches 100.

### PSP2 Process Script

<b>Purpose</b>	To guide the development of module-level programs	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Problem description</li> <li>- PSP2 Project Plan Summary form</li> <li>- Size Estimating template</li> <li>- Historical size and time data (estimated and actual)</li> <li>- Time and Defect Recording logs</li> <li>- Defect Type, Coding, and Size Measurement standards</li> <li>- Stopwatch (optional)</li> </ul>	
Step	Activities	Description
1	Planning	<ul style="list-style-type: none"> <li>- Produce or obtain a requirements statement.</li> <li>- Use the PROBE method to estimate the added and modified size of this program.</li> <li>- Complete the Size Estimating template.</li> <li>- Use the PROBE method to estimate the required development time.</li> <li>- Complete a Task Planning template.</li> <li>- Complete a Schedule Planning template.</li> <li>- Enter the plan data in the Project Plan Summary form.</li> <li>- Complete the Time Recording log.</li> </ul>
2	Development	<ul style="list-style-type: none"> <li>- Design the program.</li> <li>- <b><i>Review the design, and fix and log all defects found.</i></b></li> <li>- Implement the design.</li> <li>- <b><i>Review the code, and fix and log all defects found.</i></b></li> <li>- Compile the program, and fix and log all defects found.</li> <li>- Test the program, and fix and log all defects found.</li> <li>- Complete the Time Recording log.</li> </ul>
3	Postmortem	Complete the Project Plan Summary form with actual time, defect, and size data.
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- A thoroughly tested program</li> <li>- Completed Project Plan Summary form with estimated and actual data</li> <li>- Completed Size Estimating and Task and Schedule Planning templates</li> <li>- <b><i>Completed Design Review and Code Review checklists</i></b></li> <li>- Completed Test Report template</li> <li>- Completed PIP forms</li> <li>- Completed Time and Defect Recording logs</li> </ul>	

## PSP2 Planning Script

<b>Purpose</b>	To guide the PSP planning process	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Problem description</li> <li>- PSP2 Project Plan Summary form</li> <li>- Size Estimating, Task Planning, and Schedule Planning templates</li> <li>- Historical size and time data (estimated and actual)</li> <li>- Time Recording log</li> </ul>	
<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Program Requirements	<ul style="list-style-type: none"> <li>- Produce or obtain a requirements statement for the program.</li> <li>- Ensure that the requirements statement is clear and unambiguous.</li> <li>- Resolve any questions.</li> </ul>
2	Size Estimate	<ul style="list-style-type: none"> <li>- Produce a program conceptual design.</li> <li>- Use the PROBE method to estimate the added and modified size of this program.</li> <li>- Complete the Size Estimating template and Project Plan Summary form.</li> </ul>
3	Resource Estimate	<ul style="list-style-type: none"> <li>- Use the PROBE method to estimate the time required to develop this program.</li> <li>- Using the <i>To Date %</i> from the most recently developed program as a guide, distribute the development time over the planned project phases.</li> </ul>
4	Task and Schedule Planning	For projects lasting several days or more, complete the Task Planning and Schedule Planning templates.
5	<b>Defect Estimate</b>	<ul style="list-style-type: none"> <li>- <b><i>Based on your to-date data on defects per added and modified size unit, estimate the total defects to be found in this program.</i></b></li> <li>- <b><i>Based on your To Date % data, estimate the number of defects to be injected and removed by phase.</i></b></li> </ul>
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- Documented requirements statement</li> <li>- Program conceptual design</li> <li>- Completed Size Estimating template</li> <li>- For projects lasting several days or more, completed Task and Schedule Planning templates</li> <li>- Completed Project Plan Summary form with estimated program size, development time, <b><i>and defect</i></b> data</li> <li>- Completed Time Recording log</li> </ul>	

## PSP2 Development Script

<b>Purpose</b>	To guide the development of small programs	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Requirements statement</li> <li>- Project Plan Summary form with estimated program size and development time</li> <li>- For projects lasting several days or more, completed Task Planning and Schedule Planning templates</li> <li>- Time and Defect Recording logs</li> <li>- Defect Type standard and Coding standard</li> </ul>	
<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Design	<ul style="list-style-type: none"> <li>- Review the requirements and produce a design to meet them.</li> <li>- Record in the Defect Recording log any requirements defects found.</li> <li>- Record time in the Time Recording log.</li> </ul>
2	<i>Design Review</i>	<ul style="list-style-type: none"> <li>- <b><i>Follow the Design Review script and checklist to review the design.</i></b></li> <li>- <b><i>Fix all defects found.</i></b></li> <li>- <b><i>Record defects in the Defect Recording log.</i></b></li> <li>- <b><i>Record time in the Time Recording log.</i></b></li> </ul>
3	Code	<ul style="list-style-type: none"> <li>- Implement the design following the Coding standard.</li> <li>- Record in the Defect Recording log any requirements or design defects found.</li> <li>- Record time in the Time Recording log.</li> </ul>
4	<i>Code Review</i>	<ul style="list-style-type: none"> <li>- <b><i>Follow the Code Review script and checklist to review the code.</i></b></li> <li>- <b><i>Fix all defects found.</i></b></li> <li>- <b><i>Record defects in the Defect Recording log.</i></b></li> <li>- <b><i>Record time in the Time Recording log.</i></b></li> </ul>
5	Compile	<ul style="list-style-type: none"> <li>- Compile the program until there are no compile errors.</li> <li>- Fix all defects found.</li> <li>- Record defects in the Defect Recording log.</li> <li>- Record time in the Time Recording log.</li> </ul>
6	Test	<ul style="list-style-type: none"> <li>- Test until all tests run without error.</li> <li>- Fix all defects found.</li> <li>- Record defects in the Defect Recording log.</li> <li>- Record time in the Time Recording log.</li> <li>- Complete a Test Report template on the tests conducted and the results obtained.</li> </ul>
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- A thoroughly tested program that conforms to the Coding standard</li> <li>- <b><i>Completed Design Review and Code Review checklists</i></b></li> <li>- Completed Test Report template</li> <li>- Completed Time and Defect Recording logs</li> </ul>	

**PSP2 Design Review Script**

<b>Purpose</b>	To guide you in reviewing detailed designs	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Completed program design</li> <li>- Design Review checklist</li> <li>- Design standard</li> <li>- Defect Type standard</li> <li>- Time and Defect Recording logs</li> </ul>	
<b>General</b>	<p>Where the design was previously verified, check that the analyses</p> <ul style="list-style-type: none"> <li>- covered all of the design</li> <li>- were updated for all design changes</li> <li>- are correct</li> <li>- are clear and complete</li> </ul>	
<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Preparation	Examine the program and checklist and decide on a review strategy.
2	Review	<ul style="list-style-type: none"> <li>- Follow the Design Review checklist.</li> <li>- Review the entire program for each checklist category; do not try to review for more than one category at a time!</li> <li>- Check off each item as you complete it.</li> <li>- Complete a separate checklist for each product or product segment reviewed.</li> </ul>
3	Fix Check	<ul style="list-style-type: none"> <li>- Check each defect fix for correctness.</li> <li>- Re-review all changes.</li> <li>- Record any fix defects as new defects and, where you know the defective defect number, enter it in the fix defect space.</li> </ul>
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- A fully reviewed detailed design</li> <li>- One or more Design Review checklists for every design reviewed</li> <li>- All identified defects fixed and all fixes checked</li> <li>- Completed Time and Defect Recording logs</li> </ul>	

**Code Review Script**

<b>Purpose</b>	To guide you in reviewing programs	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- A completed and reviewed program design</li> <li>- Source program listing</li> <li>- Code Review checklist</li> <li>- Coding standard</li> <li>- Defect Type standard</li> <li>- Time and Defect Recording logs</li> </ul>	
<b>General</b>	Do the code review with a source-code listing; do not review on the screen!	
<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Review	<ul style="list-style-type: none"> <li>- Follow the Code Review checklist.</li> <li>- Review the entire program for each checklist category; do not try to review for more than one category at a time!</li> <li>- Check off each item as it is completed.</li> <li>- For multiple procedures or programs, complete a separate checklist for each.</li> </ul>
2	Correct	<ul style="list-style-type: none"> <li>- Correct all defects.</li> <li>- If the correction cannot be completed, abort the review and return to the prior process phase.</li> <li>- To facilitate defect analysis, record all of the data specified in the Defect Recording log instructions for every defect.</li> </ul>
3	Check	<ul style="list-style-type: none"> <li>- Check each defect fix for correctness.</li> <li>- Re-review all design changes.</li> <li>- Record any fix defects as new defects and, where you know the number of the defect with the incorrect fix, enter it in the fix defect space.</li> </ul>
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- A fully reviewed source program</li> <li>- One or more Code Review checklists for every program reviewed</li> <li>- All identified defects fixed</li> <li>- Completed Time and Defect Recording logs</li> </ul>	

## PSP2 Postmortem Script

<b>Purpose</b>	To guide the PSP postmortem process	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Problem description and requirements statement</li> <li>- Project Plan Summary form with program size, development time, <b>and defect</b> data</li> <li>- For projects lasting several days or more, completed Task Planning and Schedule Planning templates</li> <li>- Completed Test Report template</li> <li>- <b>Completed Design Review and Code Review checklists</b></li> <li>- Completed Time and Defect Recording logs</li> <li>- A tested and running program that conforms to the coding and size measurement standards</li> </ul>	
Step	Activities	Description
1	Defect Recording	<ul style="list-style-type: none"> <li>- Review the Project Plan Summary to verify that all of the defects found in each phase were recorded.</li> <li>- Using your best recollection, record any omitted defects.</li> </ul>
2	Defect Data Consistency	<ul style="list-style-type: none"> <li>- Check that the data on every defect in the Defect Recording log are accurate and complete.</li> <li>- Verify that the numbers of defects injected and removed per phase are reasonable and correct.</li> <li>- <b>Determine the process yield and verify that the value is reasonable and correct.</b></li> <li>- Using your best recollection, correct any missing or incorrect defect data.</li> </ul>
3	Size	<ul style="list-style-type: none"> <li>- Count the size of the completed program.</li> <li>- Determine the size of the base, reused, deleted, modified, added, total, added and modified, and new reusable code.</li> <li>- Enter these data in the Project Plan Summary form.</li> </ul>
4	Time	<ul style="list-style-type: none"> <li>- Review the completed Time Recording log for errors or omissions.</li> <li>- Using your best recollection, correct any missing or incomplete time data.</li> </ul>
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- A thoroughly tested program that conforms to the coding and size measurement standards</li> <li>- <b>Completed Design Review and Code Review checklists</b></li> <li>- Completed Test Report template</li> <li>- Completed Project Plan Summary form</li> <li>- Completed PIP forms describing process problems, improvement suggestions, and lessons learned</li> <li>- Completed Time and Defect Recording logs</li> </ul>	

**PSP2 Project Plan Summary**

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

<b>Summary</b>	<b>Plan</b>	<b>Actual</b>	<b>To Date</b>
Size/Hour _____	_____	_____	_____
Planned Time _____	_____	_____	_____
Actual Time _____	_____	_____	_____
CPI (Cost-Performance Index)			(Planned/Actual)
% Reuse _____	_____	_____	_____
% New Reusable _____	_____	_____	_____
<b>Test Defects/KLOC or equivalent</b> _____	_____	_____	_____
<b>Total Defects/KLOC or equivalent</b> _____	_____	_____	_____
<b>Yield %</b> _____	_____	_____	_____

<b>Program Size</b>	<b>Plan</b>	<b>Actual</b>	<b>To Date</b>
Base (B) _____	(Measured)	(Measured)	_____
Deleted (D) _____	(Estimated)	(Counted)	_____
Modified (M) _____	(Estimated)	(Counted)	_____
Added (A) _____	(A+M - M)	(T - B + D - R)	_____
Reused (R) _____	(Estimated)	(Counted)	_____
Added and Modified (A+M) _____	(Projected)	(A + M)	_____
Total Size (T) _____	(A+M + B - M - D + R)	(Measured)	_____
Total New Reusable _____	_____	_____	_____
Estimated Proxy Size (E) _____	_____	_____	_____

<b>Time in Phase (min.)</b>	<b>Plan</b>	<b>Actual</b>	<b>To Date</b>	<b>To Date %</b>
Planning _____	_____	_____	_____	_____
Design _____	_____	_____	_____	_____
<b>Design Review</b> _____	_____	_____	_____	_____
Code _____	_____	_____	_____	_____
<b>Code Review</b> _____	_____	_____	_____	_____
Compile _____	_____	_____	_____	_____
Test _____	_____	_____	_____	_____
Postmortem _____	_____	_____	_____	_____
Total _____	_____	_____	_____	_____

(continued)

**PSP2 Project Plan Summary (continued)**

Student _____	Program # _____			
<b>Defects Injected</b>	<i>Plan</i>	<i>Actual</i>	<b>To Date</b>	<b>To Date %</b>
Planning	_____	_____	_____	_____
Design	_____	_____	_____	_____
<b><i>Design Review</i></b>	_____	_____	_____	_____
Code	_____	_____	_____	_____
<b><i>Code Review</i></b>	_____	_____	_____	_____
Compile	_____	_____	_____	_____
Test	_____	_____	_____	_____
Total Development	_____	_____	_____	_____
<b>Defects Removed</b>	<i>Plan</i>	<i>Actual</i>	<b>To Date</b>	<b>To Date %</b>
Planning	_____	_____	_____	_____
Design	_____	_____	_____	_____
<b><i>Design Review</i></b>	_____	_____	_____	_____
Code	_____	_____	_____	_____
<b><i>Code Review</i></b>	_____	_____	_____	_____
Compile	_____	_____	_____	_____
Test	_____	_____	_____	_____
Total Development	_____	_____	_____	_____
After Development	_____	_____	_____	_____
<b><i>Defect Removal Efficiency</i></b>	<i>Plan</i>	<i>Actual</i>	<b><i>To Date</i></b>	
<b>Defects/Hour - Design Review</b>	_____	_____	_____	
<b>Defects/Hour - Code Review</b>	_____	_____	_____	
<b>Defects/Hour - Compile</b>	_____	_____	_____	
<b>Defects/Hour - Test</b>	_____	_____	_____	
<b>DRL (DLDR/UT)</b>	_____	_____	_____	
<b>DRL (Code Review/UT)</b>	_____	_____	_____	
<b>DRL (Compile/UT)</b>	_____	_____	_____	

### PSP2 Plan Summary Instructions

<b>Purpose</b>	To hold the plan and actual data for programs or program parts
<b>General</b>	<ul style="list-style-type: none"> <li>- Use the most appropriate size measure, either LOC or element count.</li> <li>- “To Date” is the total actual to-date values for all products developed.</li> <li>- A part could be a module, component, product, or system.</li> </ul>
<b>Header</b>	<ul style="list-style-type: none"> <li>- Enter your name and the date.</li> <li>- Enter the program name and number.</li> <li>- Enter the instructor’s name and the programming language you are using.</li> </ul>
<b>Summary</b>	<ul style="list-style-type: none"> <li>- Enter the added and modified size per hour planned, actual, and to-date.</li> <li>- Enter the planned and actual times for this program and prior programs.</li> <li>- For planned time to date, use the sum of the current planned time and the <b>to-date</b> planned time for the <b>most recent</b> prior program.</li> <li>- CPI = (To Date Planned Time)/(To Date Actual Time).</li> <li>- Reused % is reused size as a percentage of total program size.</li> <li>- New Reusable % is new reusable size as a percentage of added and modified size.</li> <li>- <b>Enter the test and total defects/KLOC or other appropriate measure.</b></li> <li>- <b>Enter the planned, actual, and to-date yield before compile.</b></li> </ul>
<b>Program Size</b>	<ul style="list-style-type: none"> <li>- Enter plan base, deleted, modified, reused, new reusable, and total size from the Size Estimating template.</li> <li>- Enter the plan added and modified size value (A+M) from projected added and modified size (P) on the Size Estimating template.</li> <li>- Calculate plan added size as A+M – M.</li> <li>- Enter estimated proxy size (E) from the Size Estimating template.</li> <li>- Enter actual base, deleted, modified, reused, total, and new reusable size from the Size Estimating template.</li> <li>- Calculate actual added size as T-B+D-R and actual added and modified size as A+M.</li> <li>- Enter to-date reused, added and modified, total, and new reusable size.</li> </ul>
<b>Time in Phase</b>	<ul style="list-style-type: none"> <li>- Enter plan total time in phase from the estimated total development time on the Size Estimating template.</li> <li>- Distribute the estimated total time across the development phases according to the To Date % for the most recently developed program.</li> <li>- Enter the actual time by phase and the total time.</li> <li>- To Date: Enter the sum of the actual times for this program plus the to-date times from the most recently developed program.</li> <li>- To Date %: Enter the percentage of to-date time in each phase.</li> </ul>

(continued)

**PSP2 Plan Summary Instructions (continued)**

<b>Defects Injected</b>	<ul style="list-style-type: none"> <li>- <i>Enter the total estimated defects injected.</i></li> <li>- <i>Distribute the estimated total defects across the development phases according to the To Date % for the most recently developed program.</i></li> <li>- Enter the actual defects by phase and the total actual defects.</li> <li>- To Date: Enter the sum of the actual defects injected by phase and the to-date values for the most recent previously developed program.</li> <li>- To Date %: Enter the percentage of the to-date defects injected by phase.</li> </ul>
<b>Defects Removed</b>	<ul style="list-style-type: none"> <li>- <i>Enter the estimated total defects removed.</i></li> <li>- <i>Distribute the estimated total defects across the development phases according to the To Date % for the most recently developed program.</i></li> <li>- To Date: Enter the actual defects removed by phase plus the to-date values for the most recent previously developed program.</li> <li>- To Date %: Enter the percentage of the to-date defects removed by phase.</li> <li>- After development, record any defects subsequently found during program testing, use, reuse, or modification.</li> </ul>
<b>Defect-Removal Efficiency</b>	<ul style="list-style-type: none"> <li>- <i>Calculate and enter the defects removed per hour in design review, code review, compile, and test.</i></li> <li>- <i>For DRL, take the ratio of the review and compile rates with test.</i></li> <li>- <i>Where there were no test defects, use the to-date test defect/hour value.</i></li> </ul>

## PSP2 Design Review Checklist

Student \_\_\_\_\_ Date \_\_\_\_\_  
 Program \_\_\_\_\_ Program # \_\_\_\_\_  
 Instructor \_\_\_\_\_ Language \_\_\_\_\_

<b>Purpose</b>	To guide you in conducting an effective design review				
<b>General</b>	<ul style="list-style-type: none"> <li>- Review the entire program for each checklist category; do not attempt to review for more than one category at a time!</li> <li>- As you complete each review step, check off that item in the box at the right.</li> <li>- Complete the checklist for one program or program unit before reviewing the next.</li> </ul>				
Complete	<p>Verify that the design covers all of the applicable requirements.</p> <ul style="list-style-type: none"> <li>- All specified outputs are produced.</li> <li>- All needed inputs are furnished.</li> <li>- All required includes are stated.</li> </ul>				
External Limits	Where the design assumes or relies upon external limits, determine if behavior is correct at nominal values, at limits, and beyond limits.				
Logic	<ul style="list-style-type: none"> <li>- Verify that program sequencing is proper. Stacks, lists, and so on are in the proper order. Recursion unwinds properly.</li> <li>- Verify that all loops are properly initiated, incremented, and terminated.</li> <li>- Examine each conditional statement and verify all cases.</li> </ul>				
Internal Limits	Where the design assumes or relies upon internal limits, determine if behavior is correct at nominal values, at limits, and beyond limits.				
Special Cases	<ul style="list-style-type: none"> <li>- Check all special cases.</li> <li>- Ensure proper operation with empty, full, minimum, maximum, negative, and ero values for all variables.</li> <li>- Protect against out-of-limits, overflow, and underflow conditions.</li> <li>- Ensure “impossible” conditions are absolutely impossible.</li> <li>- Handle all possible incorrect or error conditions.</li> </ul>				
Functional Use	<ul style="list-style-type: none"> <li>- Verify that all functions, procedures, or methods are fully understood and properly used.</li> <li>- Verify that all externally referenced abstractions are precisely defined.</li> </ul>				
System Considerations	<ul style="list-style-type: none"> <li>- Verify that the program does not cause system limits to be exceeded.</li> <li>- Verify that all security-sensitive data are from trusted sources.</li> <li>- Verify that all safety conditions conform to the safety specifications.</li> </ul>				
Names	Verify that <ul style="list-style-type: none"> <li>- all special names are clear, defined, and authenticated</li> <li>- the scopes of all variables and parameters are self-evident or defined</li> <li>- all named items are used within their declared scopes</li> </ul>				
Standards	Ensure that the design conforms to all applicable design standards.				

## Code Review Checklist

Student		Date	
Program		Program #	
Instructor		Language	C++

Purpose	To guide you in conducting an effective code review				
<b>General</b>	<ul style="list-style-type: none"> <li>- Review the entire program for each checklist category; do not attempt to review for more than one category at a time!</li> <li>- As you complete each review step, check off that item in the box at the right.</li> <li>- Complete the checklist for one program or program unit before reviewing the next.</li> </ul>				
Complete	Verify that the code covers all of the design.				
Includes	Verify that the includes are complete.				
Initialization	Check variable and parameter initialization. <ul style="list-style-type: none"> <li>- at program initiation</li> <li>- at start of every loop</li> <li>- at class/function/procedure entry</li> </ul>				
Calls	Check function call formats. <ul style="list-style-type: none"> <li>- pointers</li> <li>- parameters</li> <li>- use of '&amp;' </li> </ul>				
Names	Check name spelling and use. <ul style="list-style-type: none"> <li>- Is it consistent?</li> <li>- Is it within the declared scope?</li> <li>- Do all structures and classes use '.' reference?</li> </ul>				
Strings	Check that all strings are <ul style="list-style-type: none"> <li>- identified by pointers</li> <li>- terminated by NULL</li> </ul>				
Pointers	Check that all <ul style="list-style-type: none"> <li>- pointers are initialized NULL</li> <li>- pointers are deleted only after new</li> <li>- new pointers are always deleted after use</li> </ul>				
Output Format	Check the output format. <ul style="list-style-type: none"> <li>- Line stepping is proper.</li> <li>- Spacing is proper.</li> </ul>				
( ) Pairs	Ensure that ( ) are proper and matched.				
Logic Operators	<ul style="list-style-type: none"> <li>- Verify the proper use of ==, =,   , and so on.</li> <li>- Check every logic function for ( ).</li> </ul>				
Line-by-line check	Check every line of code for <ul style="list-style-type: none"> <li>- instruction syntax</li> <li>- proper punctuation</li> </ul>				
Standards	Ensure that the code conforms to the coding standards.				
File Open and Close	Verify that all files are <ul style="list-style-type: none"> <li>- properly declared</li> <li>- opened</li> <li>- closed</li> </ul>				

### PSP2.1 Process Script

<b>Purpose</b>	To guide the development of module-level programs	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Problem description</li> <li>- PSP2.1 Project Plan Summary form</li> <li>- Size Estimating template</li> <li>- Historical size and time data (estimated and actual)</li> <li>- Time and Defect Recording logs</li> <li>- Defect Type, Coding, and Size Measurement standards</li> <li>- Stopwatch (optional)</li> </ul>	
Step	Activities	Description
1	Planning	<ul style="list-style-type: none"> <li>- Produce or obtain a requirements statement.</li> <li>- Use the PROBE method to estimate the added and modified size <b><i>and the size prediction interval</i></b> of this program.</li> <li>- Complete the Size Estimating template.</li> <li>- Use the PROBE method to estimate the required development time <b><i>and the time prediction interval</i></b>.</li> <li>- Complete a Task Planning template.</li> <li>- Complete a Schedule Planning template.</li> <li>- Enter the plan data in the Project Plan Summary form.</li> <li>- Complete the Time Recording log.</li> </ul>
2	Development	<ul style="list-style-type: none"> <li>- Design the program.</li> <li>- <b><i>Document the design in the design templates.</i></b></li> <li>- Review the design, and fix and log all defects found.</li> <li>- Implement the design.</li> <li>- Review the code, and fix and log all defects found.</li> <li>- Compile the program, and fix and log all defects found.</li> <li>- Test the program, and fix and log all defects found.</li> <li>- Complete the Time Recording log.</li> </ul>
3	Postmortem	Complete the Project Plan Summary form with actual time, defect, and size data.
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- A thoroughly tested program</li> <li>- Completed Project Plan Summary form with estimated and actual data</li> <li>- Completed Size Estimating and Task and Schedule Planning templates</li> <li>- <b><i>Completed Design templates</i></b></li> <li>- Completed Design Review and Code Review checklists</li> <li>- Completed Test Report template</li> <li>- Completed PIP forms</li> <li>- Completed Time and Defect Recording logs</li> </ul>	

### PSP2.1 Planning Script

<b>Purpose</b>	To guide the PSP planning process	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Problem description</li> <li>- PSP2.1 Project Plan Summary form</li> <li>- Size Estimating, Task Planning, and Schedule Planning templates</li> <li>- Historical size and time data (estimated and actual)</li> <li>- Time Recording log</li> </ul>	
<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Program Requirements	<ul style="list-style-type: none"> <li>- Produce or obtain a requirements statement for the program.</li> <li>- Ensure that the requirements statement is clear and unambiguous.</li> <li>- Resolve any questions.</li> </ul>
2	Size Estimate	<ul style="list-style-type: none"> <li>- Produce a program conceptual design.</li> <li>- Use the PROBE method to estimate the added and modified size of this program.</li> <li>- Complete the Size Estimating template and Project Plan Summary form.</li> <li>- <b><i>Calculate the 70% size prediction interval. (You may use a spreadsheet.)</i></b></li> </ul>
3	Resource Estimate	<ul style="list-style-type: none"> <li>- Use the PROBE method to estimate the time required to develop this program.</li> <li>- <b><i>Calculate the 70% size prediction interval. (You may use a spreadsheet.)</i></b></li> <li>- Using the <u>To Date %</u> from the most recently developed program as a guide, distribute the development time over the planned project phases.</li> </ul>
4	Task and Schedule Planning	For projects lasting several days or more, complete the Task Planning and Schedule Planning templates.
5	Defect Estimate	<ul style="list-style-type: none"> <li>- Based on your to-date data on defects per added and modified size unit, estimate the total defects to be found in this program.</li> <li>- Based on your <u>To Date %</u> data, estimate the number of defects to be injected and removed by phase.</li> </ul>
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- Documented requirements statement</li> <li>- Program conceptual design</li> <li>- Completed Size Estimating template</li> <li>- For projects lasting several days or more, completed Task and Schedule Planning templates</li> <li>- Completed Project Plan Summary form with estimated program size, development time, and defect data, <b><i>and the time and size prediction intervals</i></b></li> <li>- Completed Time Recording log</li> </ul>	

### PSP2.1 Development Script

<b>Purpose</b>	To guide the development of small programs	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Requirements statement</li> <li>- Project Plan Summary form with estimated program size and development time</li> <li>- For projects lasting several days or more, completed Task Planning and Schedule Planning templates</li> <li>- Time and Defect Recording logs</li> <li>- Defect Type standard and Coding standard</li> </ul>	
Step	Activities	Description
1	Design	<ul style="list-style-type: none"> <li>- Review the requirements and produce <b><i>an external specification to meet them.</i></b></li> <li>- <b><i>Complete Functional and Operational Specification templates to record this specification.</i></b></li> <li>- <b><i>Produce a design to meet this specification.</i></b></li> <li>- <b><i>Record the design in Functional, Operational, State, and Logic Specification templates.</i></b></li> <li>- Record in the Defect Recording log any requirements defects found.</li> <li>- Record time in the Time Recording log.</li> </ul>
2	Design Review	<ul style="list-style-type: none"> <li>- Follow the Design Review script and checklist and review the design.</li> <li>- Fix all defects found.</li> <li>- Record defects in the Defect Recording log.</li> <li>- Record time in the Time Recording log.</li> </ul>
3	Code	<ul style="list-style-type: none"> <li>- Implement the design following the Coding standard.</li> <li>- Record in the Defect Recording log any requirements or design defects found.</li> <li>- Record time in the Time Recording log.</li> </ul>
4	Code Review	<ul style="list-style-type: none"> <li>- Follow the Code Review script and checklist and review the code.</li> <li>- Fix all defects found.</li> <li>- Record defects in the Defect Recording log.</li> <li>- Record time in the Time Recording log.</li> </ul>
5	Compile	<ul style="list-style-type: none"> <li>- Compile the program until there are no compile errors.</li> <li>- Fix all defects found.</li> <li>- Record defects in the Defect Recording log.</li> <li>- Record time in the Time Recording log.</li> </ul>
6	Test	<ul style="list-style-type: none"> <li>- Test until all tests run without error.</li> <li>- Fix all defects found.</li> <li>- Record defects in the Defect Recording log.</li> <li>- Record time in the Time Recording log.</li> <li>- Complete a Test Report template on the tests conducted and the results obtained.</li> </ul>
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- A thoroughly tested program that conforms to the Coding standard</li> <li>- <b><i>Completed Design templates</i></b></li> <li>- Completed Design Review and Code Review checklists</li> <li>- Completed Test Report template</li> <li>- Completed Time and Defect Recording logs</li> </ul>	

### PSP2.1 Design Review Script

<b>Purpose</b>	To guide you in reviewing detailed designs	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Completed program design <b>documented with the PSP Design templates</b></li> <li>- Design Review checklist</li> <li>- Design standard</li> <li>- Defect Type standard</li> <li>- Time and Defect Recording logs</li> </ul>	
<b>General</b>	<p>Where the design was previously verified, check that the analyses</p> <ul style="list-style-type: none"> <li>- covered all of the design</li> <li>- were updated for all design changes</li> <li>- are correct</li> <li>- are clear and complete</li> </ul>	
<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Preparation	<ul style="list-style-type: none"> <li>- Examine the program and checklist and decide on a review strategy.</li> <li>- <b>Examine the program to identify its state machines, internal loops, and variable and system limits.</b></li> <li>- <b>Use a trace table or other analytical method to verify the correctness of the design.</b></li> </ul>
2	Review	<ul style="list-style-type: none"> <li>- Follow the Design Review checklist.</li> <li>- Review the entire program for each checklist category; do not try to review for more than one category at a time!</li> <li>- Check off each item as you complete it.</li> <li>- Complete a separate checklist for each product or product segment reviewed.</li> </ul>
3	Fix Check	<ul style="list-style-type: none"> <li>- Check each defect fix for correctness.</li> <li>- Re-review all changes.</li> <li>- Record any fix defects as new defects and, where you know the defective defect number, enter it in the fix defect space.</li> </ul>
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- A fully reviewed detailed design</li> <li>- One or more Design Review checklists for every design reviewed</li> <li>- <b>Documented design analysis results</b></li> <li>- All identified defects fixed and all fixes checked</li> <li>- Completed Time and Defect Recording logs</li> </ul>	

### PSP2.1 Postmortem Script

<b>Purpose</b>	To guide the PSP postmortem process	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Problem description and requirements statement</li> <li>- Project Plan Summary form with program size, development time, and defect data</li> <li>- For projects lasting several days or more, completed Task Planning and Schedule Planning templates</li> <li>- Completed Test Report template</li> <li>- <b>Completed Design templates</b></li> <li>- Completed Design Review and Code Review checklists</li> <li>- Completed Time and Defect Recording logs</li> <li>- A tested and running program that conforms to the coding and size measurement standards</li> </ul>	
<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Defect Recording	<ul style="list-style-type: none"> <li>- Review the Project Plan Summary to verify that all of the defects found in each phase were recorded.</li> <li>- Using your best recollection, record any omitted defects.</li> </ul>
2	Defect Data Consistency	<ul style="list-style-type: none"> <li>- Check that the data on every defect in the Defect Recording log are accurate and complete.</li> <li>- Verify that the numbers of defects injected and removed per phase are reasonable and correct.</li> <li>- Determine the process yield and verify that the value is reasonable and correct.</li> <li>- Using your best recollection, correct any missing or incorrect defect data.</li> </ul>
3	Size	<ul style="list-style-type: none"> <li>- Count the size of the completed program.</li> <li>- Determine the size of the base, reused, deleted, modified, added, total, added and modified, and new reusable code.</li> <li>- Enter these data in the Project Plan Summary form.</li> </ul>
4	Time	<ul style="list-style-type: none"> <li>- Review the completed Time Recording log for errors or omissions.</li> <li>- Using your best recollection, correct any missing or incomplete time data.</li> </ul>
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- A thoroughly tested program that conforms to the coding and size measurement standards</li> <li>- <b>Completed Design templates</b></li> <li>- Completed Design Review and Code Review checklists</li> <li>- Completed Test Report template</li> <li>- Completed Project Plan Summary form</li> <li>- Completed PIP forms describing process problems, improvement suggestions, and lessons learned</li> <li>- Completed Time and Defect Recording logs</li> </ul>	

## PSP2.1 Project Plan Summary

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

<b>Summary</b>	<b>Plan</b>	<b>Actual</b>	<b>To Date</b>
Size/Hour _____	_____	_____	_____
Planned Time _____	_____	_____	_____
Actual Time _____	_____	_____	_____
CPI (Cost-Performance Index) _____	_____	_____	(Planned/Actual)
% Reuse _____	_____	_____	_____
% New Reusable _____	_____	_____	_____
Test Defects/KLOC or equivalent _____	_____	_____	_____
Total Defects/KLOC or equivalent _____	_____	_____	_____
Yield % _____	_____	_____	_____
<b>% Appraisal COQ</b> _____	_____	_____	_____
<b>% Failure COQ</b> _____	_____	_____	_____
<b>COQ A/F Ratio</b> _____	_____	_____	_____
<b>PQI</b> _____	_____	_____	_____

<b>Program Size</b>	<b>Plan</b>	<b>Actual</b>	<b>To Date</b>
Base (B) _____	(Measured)	(Measured)	_____
Deleted (D) _____	(Estimated)	(Counted)	_____
Modified (M) _____	(Estimated)	(Counted)	_____
Added (A) _____	(A+M - M)	(T - B + D - R)	_____
Reused (R) _____	(Estimated)	(Counted)	_____
Added and Modified (A+M) _____	(Projected)	(A + M)	_____
Total Size (T) _____	(A+M + B - M - D + R)	(Measured)	_____
Total New Reusable _____	_____	_____	_____
Estimated Proxy Size (E) _____	_____	_____	_____
<b>Upper Prediction Interval (70%)</b> _____	_____	_____	_____
<b>Lower Prediction Interval (70%)</b> _____	_____	_____	_____

**(continued)**

**PSP2.1 Project Plan Summary (continued)**

Student _____	Program # _____			
<b>Time in Phase (min.)</b>	<b>Plan</b>	<b>Actual</b>	<b>To Date</b>	<b>To Date %</b>
Planning	_____	_____	_____	_____
Design	_____	_____	_____	_____
Design Review	_____	_____	_____	_____
Code	_____	_____	_____	_____
Code Review	_____	_____	_____	_____
Compile	_____	_____	_____	_____
Test	_____	_____	_____	_____
Postmortem	_____	_____	_____	_____
Total	_____	_____	_____	_____
<b>Total Time UPI (70%)</b>	_____	_____	_____	_____
<b>Total Time LPI (70%)</b>	_____	_____	_____	_____
<b>Defects Injected</b>	<b>Plan</b>	<b>Actual</b>	<b>To Date</b>	<b>To Date %</b>
Planning	_____	_____	_____	_____
Design	_____	_____	_____	_____
Design Review	_____	_____	_____	_____
Code	_____	_____	_____	_____
Code Review	_____	_____	_____	_____
Compile	_____	_____	_____	_____
Test	_____	_____	_____	_____
Total Development	_____	_____	_____	_____
<b>Defects Removed</b>	<b>Plan</b>	<b>Actual</b>	<b>To Date</b>	<b>To Date %</b>
Planning	_____	_____	_____	_____
Design	_____	_____	_____	_____
Design Review	_____	_____	_____	_____
Code	_____	_____	_____	_____
Code Review	_____	_____	_____	_____
Compile	_____	_____	_____	_____
Test	_____	_____	_____	_____
Total Development	_____	_____	_____	_____
After Development	_____	_____	_____	_____
<b>Defect Removal Efficiency</b>	<b>Plan</b>	<b>Actual</b>	<b>To Date</b>	
Defects/Hour - Design Review	_____	_____	_____	
Defects/Hour - Code Review	_____	_____	_____	
Defects/Hour - Compile	_____	_____	_____	
Defects/Hour - Test	_____	_____	_____	
DRL (DLDR/UT)	_____	_____	_____	
DRL (Code Review/UT)	_____	_____	_____	
DRL (Compile/UT)	_____	_____	_____	

### PSP2.1 Plan Summary Instructions

<b>Purpose</b>	To hold the plan and actual data for programs or program parts
<b>General</b>	<ul style="list-style-type: none"> <li>- Use the most appropriate size measure, either LOC or element count.</li> <li>- “To Date” is the total actual to-date values for all products developed.</li> <li>- A part could be a module, component, product, or system.</li> </ul>
<b>Header</b>	<ul style="list-style-type: none"> <li>- Enter your name and the date.</li> <li>- Enter the program name and number.</li> <li>- Enter the instructor’s name and the programming language you are using.</li> </ul>
<b>Summary</b>	<ul style="list-style-type: none"> <li>- Enter the added and modified size per hour planned, actual, and to-date.</li> <li>- Enter the planned and actual times for this program and prior programs.</li> <li>- For planned time to date, use the sum of the current planned time and the to-date planned time for the most recent prior program.</li> <li>- CPI = (To Date Planned Time)/(To Date Actual Time).</li> <li>- Reuse % is reused size as a percentage of total program size.</li> <li>- New Reusable % is new reusable size as a percentage of added and modified size.</li> <li>- Enter the test and total defects/KLOC or other appropriate measure.</li> <li>- Enter the planned, actual, and to-date yield before compile.</li> </ul>
<b>Quality Indicators</b>	<ul style="list-style-type: none"> <li>- <i>Appraisal COQ: the percentage of development time in reviews.</i></li> <li>- <i>Failure COQ: the percentage of development time in compile and test.</i></li> <li>- <i>A/FR: the ratio of appraisal to failure COQ.</i></li> <li>- <i>Enter the planned, actual, and to-date PQI (the process quality index)</i></li> </ul>
<b>Program Size</b>	<ul style="list-style-type: none"> <li>- Enter plan base, deleted, modified, reused, new reusable, and total size from the Size Estimating template.</li> <li>- Enter the plan added and modified size value (A+M) from projected added and modified size (P) on the Size Estimating template.</li> <li>- Calculate plan added size as A+M – M.</li> <li>- Enter estimated proxy size (E) from the Size Estimating template.</li> <li>- Enter actual base, deleted, modified, reused, total, and new reusable size from the Size Estimating template.</li> <li>- Calculate actual added size as T-B+D-R and actual added and modified size as A+M.</li> <li>- Enter to-date reused, added and modified, total, and new reusable size.</li> </ul>
<b>Time in Phase</b>	<ul style="list-style-type: none"> <li>- Enter plan total time in phase from the estimated total development time on the Size Estimating template.</li> <li>- Distribute the estimated total time across the development phases according to the To Date % for the most recently developed program.</li> <li>- Enter the actual time by phase and the total time.</li> <li>- To Date: Enter the sum of the actual times for this program plus the to-date times from the most recently developed program.</li> <li>- To Date %: Enter the percentage of to-date time in each phase.</li> </ul>
<b>Prediction Interval</b>	<ul style="list-style-type: none"> <li>- <i>Enter the 70% UPI and LPI total size and time ranges.</i></li> </ul>

(continued)

**PSP2.1 Plan Summary Instructions (continued)**

<b>Defects Injected</b>	<ul style="list-style-type: none"> <li>- Enter the total estimated defects injected.</li> <li>- Distribute the estimated total defects across the development phases according to the To Date % for the most recently developed program.</li> <li>- Enter the actual defects by phase and the total actual defects.</li> <li>- To Date: Enter the sum of the actual defects injected by phase and the to-date values for the most recent previously developed program.</li> <li>- To Date %: Enter the percentage of the to-date defects injected by phase.</li> </ul>
<b>Defects Removed</b>	<ul style="list-style-type: none"> <li>- Enter the estimated total defects removed.</li> <li>- Distribute the estimated total defects across the development phases according to the To Date % for the most recently developed program.</li> <li>- To Date: Enter the actual defects removed by phase plus the To Date values for the most recent previously developed program.</li> <li>- To Date %: Enter the percentage of the To Date defects removed by phase.</li> <li>- After development, record any defects subsequently found during program testing, use, reuse, or modification.</li> </ul>
<b>Defect-Removal Efficiency</b>	<ul style="list-style-type: none"> <li>- Calculate and enter the defects removed per hour in design review, code review, compile, and test.</li> <li>- For DRL, take the ratio of the review and compile rates with test.</li> <li>- Where there were no test defects, use the to-date test defect/hour value.</li> </ul>

## Design Review Checklist

Student \_\_\_\_\_ Date \_\_\_\_\_  
 Program \_\_\_\_\_ Program # \_\_\_\_\_  
 Instructor \_\_\_\_\_ Language \_\_\_\_\_

Purpose	To guide you in conducting an effective design review			
<b>General</b>	<ul style="list-style-type: none"> <li>- Review the entire program for each checklist category; do not attempt to review for more than one category at a time!</li> <li>- As you complete each review step, check off that item in the box at the right.</li> <li>- Complete the checklist for one program or program unit before reviewing the next.</li> </ul>			
Complete	<p>Verify that the design covers all of the applicable requirements.</p> <ul style="list-style-type: none"> <li>- All specified outputs are produced.</li> <li>- All needed inputs are furnished.</li> <li>- All required includes are stated.</li> </ul>			
External Limits	Where the design assumes or relies upon external limits, determine if behavior is correct at nominal values, at limits, and beyond limits.			
Logic	<p><b><i>Use a trace table, mathematical proof, or similar method to verify the logic.</i></b></p> <ul style="list-style-type: none"> <li>- Verify that program sequencing is proper. Stacks, lists, and so on are in the proper order. Recursion unwinds properly.</li> <li>- Verify that all loops are properly initiated, incremented, and terminated.</li> <li>- Examine each conditional statement and verify all cases.</li> </ul>			
<b>State Analysis</b>	<b><i>For each state machine, verify that the state transitions are all complete and orthogonal.</i></b>			
Internal Limits	Where the design assumes or relies upon internal limits, determine if behavior is correct at nominal values, at limits, and beyond limits.			
Special Cases	<ul style="list-style-type: none"> <li>- Check all special cases.</li> <li>- Ensure proper operation with empty, full, minimum, maximum, negative, and zero values for all variables.</li> <li>- Protect against out-of-limits, overflow, and underflow conditions.</li> <li>- Ensure “impossible” conditions are absolutely impossible.</li> <li>- Handle all possible incorrect or error conditions.</li> </ul>			
Functional Use	<ul style="list-style-type: none"> <li>- Verify that all functions, procedures, or methods are fully understood and properly used.</li> <li>- Verify that all externally referenced abstractions are precisely defined.</li> </ul>			
System Considerations	<ul style="list-style-type: none"> <li>- Verify that the program does not cause system limits to be exceeded.</li> <li>- Verify that all security-sensitive data are from trusted sources.</li> <li>- Verify that all safety conditions conform to the safety specifications.</li> </ul>			
Names	<p>Verify that</p> <ul style="list-style-type: none"> <li>- all special names are clear, defined, and authenticated</li> <li>- the scopes of all variables and parameters are self-evident or defined</li> <li>- all named items are used within their declared scopes</li> </ul>			
Standards	Ensure that the design conforms to all applicable design standards.			

# **Operational Specification Template**

**Student** \_\_\_\_\_ **Date** \_\_\_\_\_  
**Program** \_\_\_\_\_ **Program #** \_\_\_\_\_  
**Instructor** \_\_\_\_\_ **Language** \_\_\_\_\_

**Operational** Specification Template Instructions

<b>Purpose</b>	<ul style="list-style-type: none"> <li>- To hold descriptions of the likely operational scenarios followed during program use</li> <li>- To ensure that all significant usage issues are considered during program design</li> <li>- To specify test scenarios</li> </ul>
<b>General</b>	<ul style="list-style-type: none"> <li>- Use this template for complete programs, subsystems, or systems.</li> <li>- Group multiple small scenarios on a single template, as long as they are clearly distinguished and have related objectives.</li> <li>- List the major scenarios and reference other exception, error, or special cases under comments.</li> <li>- Use this template to document the operational specifications during planning, design, test development, implementation, and test.</li> <li>- After implementation and testing, update the template to reflect the actual implemented product.</li> </ul>
<b>Header</b>	<ul style="list-style-type: none"> <li>- Enter your name and the date.</li> <li>- Enter the program name and number.</li> <li>- Enter the instructor's name and the programming language you are using.</li> </ul>
<b>Scenario Number</b>	Where several scenarios are involved, reference numbers are needed.
<b>User Objective</b>	List the users' likely purpose for the scenario, for example, to log onto the system or to handle an error condition.
<b>Scenario Objective</b>	List the designer's purpose for the scenario, for example, to define common user errors or to detail a test scenario.
<b>Source</b>	<ul style="list-style-type: none"> <li>- Enter the source of the scenario action.</li> <li>- Example sources could be user, program, and system.</li> </ul>
<b>Step</b>	Provide sequence numbers for the scenario steps. These facilitate reviews and inspections.
<b>Action</b>	<p>Describe the action taken, such as</p> <ul style="list-style-type: none"> <li>- Enter incorrect mode selection.</li> <li>- Provide error message.</li> </ul>
<b>Comments</b>	<p>List significant information relating to the action, such as</p> <ul style="list-style-type: none"> <li>- User enters an incorrect value.</li> <li>- An error is possible with this action.</li> </ul>

**Functional Specification Template**

<b>Student</b>	_____	<b>Date</b>	_____
<b>Program</b>	_____	<b>Program #</b>	_____
<b>Instructor</b>	_____	<b>Language</b>	_____

<b>Class Name</b>	
<b>Parent Class</b>	

<b>Attributes</b>	
<b>Declaration</b>	<b>Description</b>

<b>Items</b>	
<b>Declaration</b>	<b>Description</b>

### Functional Specification Template Instructions

<b>Purpose</b>	<ul style="list-style-type: none"> <li>- To hold a part's functional specifications</li> <li>- To describe classes, program modules, or entire programs</li> </ul>
<b>General</b>	<ul style="list-style-type: none"> <li>- Use this template for complete programs, subsystems, or systems.</li> <li>- Use this template to document the functional specifications during planning, design, test development, implementation, and test.</li> <li>- After implementation and testing, update the template to reflect the actual implemented product.</li> </ul>
<b>Header</b>	<ul style="list-style-type: none"> <li>- Enter your name and the date.</li> <li>- Enter the program name and number.</li> <li>- Enter the instructor's name and the programming language you are using.</li> </ul>
<b>Class Name</b>	<ul style="list-style-type: none"> <li>- Enter the part or class name and the classes from which it directly inherits.</li> <li>- List the class names starting with the most immediate.</li> <li>- Where practical, list the full inheritance hierarchy.</li> </ul>
<b>Attributes</b>	<ul style="list-style-type: none"> <li>- Provide the declaration and description for each global or externally visible variable or parameter with any constraints.</li> <li>- List pertinent relationships of this part with other parts together with the multiplicity and constraints.</li> </ul>
<b>Items</b>	<ul style="list-style-type: none"> <li>- Provide the declaration and description for each item.</li> <li>- Precisely describe the conditions that govern each item's return values.</li> <li>- Describe any initialization or other key item responsibilities.</li> </ul>
<b>Example Items</b>	An item could be a class method, procedure, function, or database query, for example.

## State Specification Template

Student \_\_\_\_\_ Date \_\_\_\_\_  
Program \_\_\_\_\_ Program # \_\_\_\_\_  
Instructor \_\_\_\_\_ Language \_\_\_\_\_

### State Specification Template Instructions

<b>Purpose</b>	<ul style="list-style-type: none"> <li>- To hold the state and state transition specifications for a system, class, or program</li> <li>- To support state-machine analysis during design, design reviews, and design inspections</li> </ul>
<b>General</b>	<ul style="list-style-type: none"> <li>- This form shows each system, program, or routine state, the attributes of that state, and the transition conditions among the states.</li> <li>- Use this template to document the state specifications during planning, design, test development, implementation, and test.</li> <li>- After implementation and testing, update the template to reflect the actual implemented product.</li> </ul>
<b>Header</b>	<ul style="list-style-type: none"> <li>- Enter your name and the date.</li> <li>- Enter the program name and number.</li> <li>- Enter the instructor's name and the programming language you are using.</li> </ul>
<b>State Name</b>	<ul style="list-style-type: none"> <li>- Name all of the program's states.</li> <li>- Also enter each state name in the header space at the top of each "States/Next States" section of the template.</li> </ul>
<b>State Name Description</b>	<ul style="list-style-type: none"> <li>- Describe each state and any parameter values that characterize it.</li> <li>- For example, if a state is described by SetSize=10 and SetPosition=3, list SetSize=10 and SetPosition=3.</li> </ul>
<b>Function/Parameter</b>	<ul style="list-style-type: none"> <li>- List the principal functions and parameters.</li> <li>- Include all key variables or methods used to define state transitions or actions.</li> </ul>
<b>Function/Parameter Description</b>	<ul style="list-style-type: none"> <li>- For each function, provide its declaration, parameters, and returns.</li> <li>- For each parameter, define its type and significant values.</li> </ul>
<b>Next State</b>	<ul style="list-style-type: none"> <li>- For each state, list the names of all possible next states.</li> <li>- Include the state itself.</li> </ul>
<b>Transition Condition</b>	<p>List the conditions for transition to each next state.</p> <ul style="list-style-type: none"> <li>- Use a mathematical or otherwise precise notation.</li> <li>- If the transition is impossible, list "impossible," with a note saying why.</li> </ul>
<b>Action</b>	<p>List the actions taken with each state transition.</p>

## Logic Specification Template

Student \_\_\_\_\_ Date \_\_\_\_\_  
Program \_\_\_\_\_ Program # \_\_\_\_\_  
Instructor \_\_\_\_\_ Language \_\_\_\_\_

---

**Design**    [References](#)

<b>Parameters</b>	<hr/> <hr/> <hr/> <hr/> <hr/>
-------------------	-------------------------------

### Logic Specification Template Instructions

<b>Purpose</b>	<ul style="list-style-type: none"> <li>- To contain the pseudocode for a program, component, or system</li> <li>- To enable precise and complete program implementation</li> <li>- To facilitate thorough design and implementation reviews and inspections</li> </ul>
<b>General</b>	<ul style="list-style-type: none"> <li>- Use this template to document the program's detailed logic.</li> <li>- After implementation and testing, update the template to reflect the actual implemented product.</li> <li>- During detailed design, write the pseudocode needed to describe all of the program's logic.</li> <li>- Use plain language and avoid using programming instructions wherever practical.</li> </ul>
<b>Header</b>	<ul style="list-style-type: none"> <li>- Enter your name and the date.</li> <li>- Enter the program name and number.</li> <li>- Enter the instructor's name and the programming language you are using.</li> </ul>
<b>Design References</b>	<p>List the references used to produce the program's logical design.</p> <ul style="list-style-type: none"> <li>- the Operational, Functional, and State templates</li> <li>- the program's requirements</li> <li>- any other pertinent source</li> </ul>
<b>Parameters</b>	<ul style="list-style-type: none"> <li>- Where needed, define any parameters or abbreviations used.</li> <li>- Avoid duplicating definitions on other templates and reference these other definitions where they are needed.</li> </ul>

## Expanded Defect Type Standard

<b>Purpose</b>		To facilitate causal analysis and defect prevention	
<b>Note</b>		The types are grouped in ten general categories. - If the detailed category does not apply, use the general category. - The % column lists an example type distribution.	
No.	Name	Description	%
10	Documentation	Comments, messages, manuals	1.1
20	Syntax	General syntax problems	0.8
21	Typos	Spelling, punctuation	32.1
22	Instruction formats	General format problems	5.0
23	Begin-end	Did not properly delimit operation	0
30	Packaging	Change management, version control, system build	1.6
40	Assignment	General assignment problem	0
41	Naming	Declaration, duplicates	12.6
42	Scope		1.3
43	Initialize and close	Variables, objects, classes, and so on	4.0
44	Range	Variable limits, array range	0.3
50	Interface	General interface problems	1.3
51	Internal	Procedure calls and references	9.5
52	I/O	File, display, printer, communication	2.6
53	User	Formats, content	8.9
60	Checking	Error messages, inadequate checks	0
70	Data	Structure, content	0.5
80	Function	General logic	1.8
81	Pointers	Pointers, strings	8.7
82	Loops	Off-by-one, incrementing, recursion	5.5
83	Application	Computation, algorithmic	2.1
90	System	Timing, memory, and so on	0.3
100	Environment	Design, compile, test, other support system problems	0

### The Process Development Process Script – Script PDP

<b>Purpose</b>	- To guide process development or enhancement
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- A new or updated process is needed.</li> <li>- Requirements or PIPs are available describing the process needs.</li> <li>- The PDP forms, scripts, standards, and review checklist are on hand.</li> </ul>
<b>General</b>	<ul style="list-style-type: none"> <li>- The PDP process may be followed by one developer or an entire team.</li> <li>- When used by teams, PDP is incorporated into the team process.</li> <li>- Develop the scripts, forms, standards, and other elements in that order.</li> <li>- Break larger processes into sections and use PDP to develop each section.</li> </ul>

<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Produce the Process Plan	<ul style="list-style-type: none"> <li>- Inventory the existing process (if any).</li> <li>- Produce the new or modified process conceptual design.</li> <li>- Estimate the new, reused, modified, and deleted process elements.</li> <li>- Separately estimate the development effort for the scripts, forms, standards, support materials, and other process elements.</li> <li>- Estimate the total time required for each process element type.</li> <li>- Estimate the total time required for each process phase.</li> <li>- Complete the PDP Plan Summary.</li> </ul>
2	Process Requirements	<ul style="list-style-type: none"> <li>- Analyze the PIPs or other requirements documents.</li> <li>- Define the requirements for the new or modified process.</li> </ul>
3	High-level Design	<p>For new or modified processes</p> <ul style="list-style-type: none"> <li>- Define any process constraints such as training, special tools, or support.</li> <li>- Define the process flow and principal phases.</li> </ul>
4	Process Scripts	<ul style="list-style-type: none"> <li>- Produce or modify each process script.</li> <li>- Using the review checklist, review and correct each process script.</li> </ul>
5	Process Forms	<ul style="list-style-type: none"> <li>- Produce or modify each process form.</li> <li>- Using the review checklist, review and correct each process form.</li> </ul>
6	Process Standards	<ul style="list-style-type: none"> <li>- Produce or modify each process standard.</li> <li>- Using the review checklist, review and correct each process standard.</li> </ul>
7	Other Process Elements	<ul style="list-style-type: none"> <li>- Produce or modify the other process elements.</li> <li>- Using the review checklist, review and correct the other process elements.</li> </ul>
8	Process Notebook	<ul style="list-style-type: none"> <li>- Produce or modify the process notebook, if required.</li> <li>- Using the review checklist, review and correct the process notebook.</li> </ul>
9	Process Test	<ul style="list-style-type: none"> <li>- Test the process by following the script to complete all of the forms.</li> <li>- Where possible, use the process on a real project.</li> <li>- Record all problems and improvement suggestions on PIPs.</li> <li>- Adjust the process to address all of the PIPs.</li> </ul>
10	Peer Review	<ul style="list-style-type: none"> <li>- Where possible, have peers review the process to identify errors, and suggest modifications.</li> <li>- Update the process based on the review results.</li> </ul>
11	Process Packaging	<ul style="list-style-type: none"> <li>- Package the finished process in a form suitable for distribution and use.</li> <li>- Check the final package for errors and make needed corrections.</li> </ul>
12	Postmortem	<ul style="list-style-type: none"> <li>- Record the process data on the Plan Summary.</li> <li>- Calculate the element and phase times and rates.</li> <li>- Record these data in the process database for use in future planning.</li> <li>- Note any PDP process improvements on form PIP.</li> </ul>
<b>Exit Criteria</b>		<ul style="list-style-type: none"> <li>- A completed high-quality process ready for distribution and use</li> <li>- A Project Summary report</li> <li>- Planning data for future use</li> </ul>

**PSP3.0 Development Script**

<b>Purpose</b>	To guide development of component-level programs	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Problem description or component specifications</li> <li>- Process forms and standards</li> <li>- Historical estimated and actual size and time data</li> </ul>	
<b>General</b>	<ul style="list-style-type: none"> <li>- Where the forms and templates are in a support tool, print and retain completed copies in a personal file or notebook.</li> <li>- Where suitable tool support is not available, complete and retain paper copies of all required PSP3 forms.</li> <li>- Record the time spent in each process step.</li> <li>- Record all defects found.</li> </ul>	
<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Requirements and Planning	<ul style="list-style-type: none"> <li>- Obtain the requirements and produce the development plan.           <ul style="list-style-type: none"> <li>- requirements document</li> <li>- design concept</li> <li>- size, quality, resource, and schedule plans</li> </ul> </li> <li>- Produce a master Issue Tracking log.</li> </ul>
2	High-level Design (HLD)	<p>Produce the design and implementation strategy.</p> <ul style="list-style-type: none"> <li>- Functional Specifications</li> <li>- State Specifications</li> <li>- Operational Specifications</li> <li>- development strategy</li> <li>- test strategy and plan</li> </ul>
3	High-level Design Review (HLDR)	<ul style="list-style-type: none"> <li>- Review the high-level design.</li> <li>- Review the development and test strategy.</li> <li>- Fix and log all defects found.</li> <li>- Note outstanding issues on the Issue Tracking log.</li> <li>- Log all defects found.</li> </ul>
4	Development (PSP 2.1)	<ul style="list-style-type: none"> <li>- Design the program and document the design in the PSP Design templates.</li> <li>- Review the design and fix and log all defects.</li> <li>- Implement the design.</li> <li>- Review the code and fix and log all defects.</li> <li>- Compile the program and fix and log all defects.</li> <li>- Test the program and fix and log all defects.</li> <li>- Complete the Time Recording log.</li> <li>- Reassess and recycle as needed.</li> </ul>
5	Postmortem	Complete the Project Plan Summary form with the actual time, defect, and size data.
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- A thoroughly tested program</li> <li>- Completed Project Plan Summary with estimated and actual data</li> <li>- Completed Estimating and Planning templates</li> <li>- Completed Design templates</li> <li>- Completed Design Review checklist and Code Review checklist</li> <li>- Completed Test Report template</li> <li>- Completed Issue Tracking log</li> <li>- Completed PIP forms</li> <li>- Completed Time and Defect Recording logs</li> </ul>	

### Prototype Experimental Process Script (PEP)

<b>Purpose</b>	To guide PSP experimental program development	
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Requirements statement or a problem description</li> <li>- All PEP process scripts, forms, standards, templates, and logs</li> <li>- Historical data for estimated and actual size, time, and quality</li> </ul>	
<b>General</b>	<ul style="list-style-type: none"> <li>- Where the forms and templates are in a support tool, print and retain completed copies in a personal file or notebook.</li> <li>- Where suitable tool support is not available, complete and retain paper copies of all required PSP forms.</li> <li>- Record time spent, defects found, and product size produced at each process step.</li> </ul>	
<b>Step</b>   <b>Activities</b>   <b>Description</b>		
1	Planning	<ul style="list-style-type: none"> <li>- Base development planning on the best available requirements information.</li> <li>- Produce the program's conceptual design.</li> <li>- Use the PROBE method and Size Estimating template to estimate the program size, development time, and prediction intervals.</li> <li>- For projects of more than a few days duration, complete Task and Schedule Planning templates.</li> <li>- Enter the plan data in the Project Plan and Quality Summaries.</li> </ul>
2	Requirements and Strategy	<ul style="list-style-type: none"> <li>- Assess the plan and conceptual design to identify risks and uncertainties.</li> <li>- Where the risks are significant, start with a prototype development phase.</li> </ul>
3	Prototype Development	Develop one or more working prototypes to resolve the identified risks and uncertainties.
4	Development (PSP2.1)	<ul style="list-style-type: none"> <li>- Produce and document the product's design in State, Operational, Functional, and Logic Specification templates.</li> <li>- Provide designs for all the prototype elements that will be used in the final product.</li> <li>- Review, analyze, and correct the design.</li> <li>- Plan and develop tests to verify correct program operation.</li> <li>- Where required, implement the program's design.</li> <li>- Review and correct the implementation.</li> <li>- Compile, build, test, and fix the program until all tests run without error.</li> </ul>
5	Postmortem	<ul style="list-style-type: none"> <li>- Complete the Project Plan and Quality Summaries with actual time, defect, size, and quality data.</li> <li>- Update the Design and Code Review checklists to reflect new defect data.</li> <li>- Produce PIPs for any process improvement ideas.</li> </ul>
<b>Exit Criteria</b>	<ul style="list-style-type: none"> <li>- A documented, analyzed, reviewed, tested, and corrected program</li> <li>- Project Plan and Quality Summaries with estimated and actual data</li> <li>- Completed Estimating, Planning, Design, and Test Report templates</li> <li>- Updated Design and Code Review checklists.</li> <li>- Completed Defect Recording, Time Recording, and Issue Tracking logs</li> <li>- A personal project notebook with plan and actual data</li> <li>- Completed PIP forms</li> </ul>	

### Product Maintenance Process Script (PMP)

<b>Purpose</b>	To guide PSP enhancement and repair of legacy systems
<b>Entry Criteria</b>	<ul style="list-style-type: none"> <li>- Requirements statement or a problem description</li> <li>- All PSP and PMP process scripts, forms, standards, templates, and logs</li> <li>- Historical data for estimated and actual project size, time, and quality</li> <li>- Historical data for legacy system test defects and customer-reported defects</li> <li>- defect data for every system component</li> <li>- for defective components, defect data by module</li> </ul>
<b>General</b>	<ul style="list-style-type: none"> <li>- Rank the replacement modules by level of user impact.</li> <li>- Where the forms and templates are in a support tool, print and retain completed copies in a personal file or notebook.</li> <li>- Where suitable tool support is not available, complete and retain paper copies of all required PSP and PMP forms.</li> <li>- Record time spent, defects found, and product size produced at each process step.</li> </ul>

<b>Step</b>	<b>Activities</b>	<b>Description</b>
1	Planning	<ul style="list-style-type: none"> <li>- Base development planning on the enhancement requirements and the resources allocated for cleaning up (remediating) the legacy system.</li> <li>- Produce and document the enhancements' conceptual designs.</li> <li>- Use the PROBE method and Size Estimating template to estimate the size of each enhancement, the development time, and the prediction intervals.</li> <li>- Include the allocated remediation effort in the task and schedule plans.</li> <li>- Enter the plan data in the Project Plan and Quality Summaries.</li> </ul>
2	Requirements Review	Review and inspect the enhancement requirements and resolve any open issues or questions.
3	Maintenance Strategy	<ul style="list-style-type: none"> <li>- From the enhancement requirements and conceptual design, identify the legacy components and modules to be modified.</li> <li>- Review the legacy-system defect data and classify components and component modules as defect-free, defective, and replacement.</li> <li>- Rank the replacement modules in priority order, based on user impact.</li> <li>- Identify all of the defective and replacement modules that are to be changed by the enhancements.</li> </ul>
4	Enhancement – Defect-free Modules	Follow a personal PSP process to develop, review, inspect, and test the required enhancements for each defect-free module.
5	Enhancement – Defective Modules	<p>For each defective module with one or more planned enhancements</p> <ul style="list-style-type: none"> <li>- Review the unmodified module and correct any defects found.</li> <li>- If any module has more than one defect found or has design problems, hold a team inspection and fix all defects found.</li> <li>- If the module's design problems are serious, reclassify the module for replacement and proceed to script step 6.</li> <li>- Test and regression test the fixed module before making the enhancements.</li> <li>- Follow a PSP2.1-like process to develop, review, inspect, and test the module's enhancements.</li> </ul>

**(continued)**

**Product Maintenance Process Script (PMP) (continued)**

6	Enhancement of Replacement Modules	<p>For each replacement module with one or more planned enhancements</p> <ul style="list-style-type: none"> <li>- Document the unmodified module's design with the PSP Design templates.</li> <li>- If the design has serious defects or is overly complex, produce a new design and implementation, and review and inspect the design and implementation.</li> <li>- If the design appears sound, review and fix the design and code, hold a team inspection and fix all of the defects found.</li> <li>- Test and regression test the fixed module before making the enhancements.</li> <li>- Follow a personal PSP process to develop, review, inspect, and test the module's enhancements.</li> </ul>
7	Module Remediation	<p>If remediation resources are available, start with the highest-ranked replacement module that has no planned enhancements.</p> <ul style="list-style-type: none"> <li>- Document the unmodified module's design with the PSP Design templates.</li> <li>- If the design has serious defects or is overly complex, follow a PSP2.1-like process to produce a new design and implementation and to review and inspect the design and implementation.</li> <li>- If the design appears sound, review and fix the design and code, hold a team inspection and fix all defects found.</li> <li>- Test and regression test the replacement module.</li> </ul>
5	Postmortem	<ul style="list-style-type: none"> <li>- Complete the Project Plan and Quality Summaries with actual time, defect, size, and quality data.</li> <li>- Update the Design and Code Review checklists to reflect new defect data.</li> <li>- Produce a remediation report, identifying the replaced components and modules. <ul style="list-style-type: none"> <li>- Summarize the time, size, and defect data for each module repair and replacement.</li> <li>- Update the historical database with the remediation data for use in planning future maintenance efforts.</li> </ul> </li> <li>- Produce PIPs for any process improvement ideas.</li> </ul>
<b>Exit Criteria</b>		<ul style="list-style-type: none"> <li>- Documented, analyzed, reviewed, tested, and enhanced legacy program</li> <li>- Project Plan and Quality Summaries with estimated and actual data</li> <li>- Completed Estimating, Planning, Design, and Test Report templates</li> <li>- Updated Design and Code Review checklists</li> <li>- Completed Defect Recording, Time Recording, and Issue Tracking logs</li> <li>- Personal project notebook with plan and actual data</li> <li>- Maintenance report and updated database of remediation data</li> <li>- Completed PIP forms</li> </ul>