

Categories	Level 1 (50-59%)	Level 2 (60 - 69%)	Level 3 (70 - 79%)	Level 4 (80 - 100%)
Program Logic and Design <ul style="list-style-type: none"> • use of appropriate control structures (eg. if-else, case, loops...) 	inappropriate choice and use of control structures	occasional choice and use of appropriate control structures	frequent choice and use of appropriate control structures	consistent choice and use of appropriate control structures
<ul style="list-style-type: none"> • modular design (units, procedures, functions) 	inappropriate or limited use of modular design	occasional appropriate use of modular design	consistent appropriate use of modular design	extensive advanced use of modular design
<ul style="list-style-type: none"> • program easy to modify (eg. use of const, types..) 	modifications require extensive program changes	modifications require minor program changes	program relatively easy to modify	program easy to modify quickly and efficiently
Program Execution and Usage <ul style="list-style-type: none"> • successful program execution (normal and special input cases) 	program runs partially, or runs mostly incorrectly for normal input	program runs successfully for most normal input cases	program runs successfully and correctly for normal input cases	program runs successfully and correctly for normal & special input
<ul style="list-style-type: none"> • user friendliness (error, I/O messages) 	apparent lack of I/O and error messages	occasional use of I/O and error messages	frequent use of I/O and error messages	consistent accurate use of I/O & error messages
Program Specifications <ul style="list-style-type: none"> • program meets assignment specifications (language features, I/O requirements) 	program satisfies few assignment specifications	program satisfies some of the assignment specifications, but many still not met	program satisfies the assignment specifications mostly correctly	program satisfies the assignment specifications completely and correctly
Coding Style <ul style="list-style-type: none"> • logical arrangement of code structure (modules, type, var, const..) 	apparent lack of logical and consistent organization of code structure	consistent but inappropriate organization of code structure	mostly consistent and logical organization of code structure	consistent, logical organization of code structure applied throughout
<ul style="list-style-type: none"> • proper code indentation techniques 	code indentation inconsistent and inappropriate	consistent but inappropriate code indentation	mostly consistent, appropriate code indentation	complete, consistent, appropriate indentation
<ul style="list-style-type: none"> • appropriate identifier names (procs, vars..) 	inappropriate names used, but consistently	descriptive names used inconsistently	descriptive names used relatively consistently	descriptive names used consistently throughout
Documentation <ul style="list-style-type: none"> • detailed program header (title, purpose, course code, etc.) 	inappropriate program header or no header used	program header apparent but limited in detail	accurate, somewhat detailed program header	accurate, thoroughly detailed program header
<ul style="list-style-type: none"> • program comments (vars, consts, types, program modules..) 	no comments or very limited comments apparent	some limited, inappropriate or inaccurate comments apparent	comments apparent throughout but lack in detail or accuracy	thorough, accurate comments used consistently throughout
Creativity <ul style="list-style-type: none"> • user interface (design, colour, interactivity, user-interest, creativity) 	minimal interactivity, crude interface design	somewhat interactive and/or creative interface design	interesting, interactive, user-friendly interface design	highly advanced, interactive, user-friendly interface design
<ul style="list-style-type: none"> • language features and concept complexity (standard and advanced) 	no advanced and lack of standard language features and concepts used	standard language features and concepts used correctly	some advanced language features and/or complex concepts used correctly	highly complex concept and advanced language features used correctly