***Procedural Programming 1: Marking Rubric***

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | ***Criteria & Outcomes***  ***Total Marks: 10*** |
| ***Not***  ***Included***  ***0*** | ***Partially***  ***Met***  ***1*** | ***Fully***  ***Met***  ***2*** | **Procedural Elements Research Assignment: 4 Marks** |
|  |  |  | **Knowledge: demonstrate an understanding of modular programming**  1.1 describe the advantages of programming with modules or subroutines including:  1.1.1 reducing the duplication of code in a program  1.1.2 enabling the reuse of code in more than one program  1.1.3 decomposing complex problems into simpler pieces to improve maintainability and  extendibility  1.1.4 improving the readability of a program  1.1.5 hiding or protecting the program data  1.2 select a programming environment and describe how it supports procedural programming  including:  1.2.1 the type of subprograms supported; e.g., procedures, functions, methods  1.2.2 the level or type of modularity provided  1.2.3 the level of protection provided from unwanted side-effects |
|  |  |  | **Life Roles: identify possible life roles related to the skills and content of this cluster**  6.1 recognize and then analyze the opportunities and barriers in the immediate environment  6.2 identify potential resources to minimize barriers and maximize opportunities |
| ***Not***  ***Included***  ***0*** | ***Partially***  ***Met***  ***1*** | ***Fully***  ***Met***  ***2*** | **Number Guesser Assignment: 6 Marks** |
|  |  |  | **Programming:** **demonstrate basic procedural programming skills by writing algorithms employing a modular**  **approach to solve problems**  2.1 analyze a data processing problem and use a top-down design approach to decompose it into  discreet input, processing and output modules  2.2 analyze and refine modules into submodules that are a manageable size for each process; e.g.,  input submodules, processing submodules and output submodules  2.3 describe and represent, using pseudocode or an appropriate diagramming approach, the  relationship among the modules  2.4 analyze and rewrite algorithms for each module identifying the pre- and post-conditions and  required program control of flow mechanisms.  2.5 analyze and evaluate algorithms for each developing module with appropriate data and revise, as  required |
|  |  |  | **Modify/Debug: compare the results of the program with the intent of the algorithm and modify, as required**  4.1 use appropriate error trapping mechanisms built into the programming environment, as well as  programmer-directed error-trapping techniques, to eliminate logic errors and debug the program  4.2 compare the congruency between the outcomes of the debugged program and the original intent  of the algorithm and modify, as required |
|  |  |  | **Basic Competencies: demonstrate basic competencies**  5.1 demonstrate fundamental skills to:  5.1.1 communicate  5.1.2 manage information  5.1.3 use numbers  5.1.4 think and solve problems  5.2 demonstrate personal management skills to:  5.2.1 demonstrate positive attitudes and behaviours  5.2.2 be responsible  5.2.3 be adaptable  5.2.4 learn continuously  5.2.5 work safely  5.3 demonstrate teamwork skills to:  5.3.1 work with others  5.3.2 participate in projects and tasks |