GUJARAT TECHNOLOGICAL UNIVERSITY

Master in Computer Application (Integrated MCA)

Year II – (Semester-IV) (W.E.F. January 2015)

Subject Name: Software Project-4 (SP4)

Subject Code: 4440605

Guidelines for evaluation of practical performance of students in the examination of subject

Project in C++

These guidelines can be shared with students to help them improve and know in a better what is to be performed and why is it actually needed

1. The students should be evaluated on the basis of what they know rather than what they do not know. The program should exhibit student's knowledge of C++ and not the domain in which he has worked. Thus the coding ability of the student should take preference over their domain knowledge. **The entire project should be evaluated on the basis of the code programmed and if required changes/modifications should be given to the student in his project. [Documentation and presentation are not mandatory. The prime parameter of evaluation should be the code only]**
2. The system developed by students can be evaluated on the basis of following suggested points:
    a. All the programs must be strictly developed in C++ language only
    b. Check should be made that the concepts required in the problem are correctly implemented. This can be done checking answers to following questions:
        i. Are there well defined classes?
        ii. Are all the possible constructors and destructors provided in almost all classes designed? If not provided then what is the important reason of not providing?

iii. Is MIL used instead of assignment whenever possible?

iv. Proper level of abstraction is done or not? That means, is the class design indicating the level of abstraction required or not? Ideally all entities represented as appropriate classes with required attributes as members.

v. Are member variables defined public/private/protected as required or not?

vi. Are operators overloaded to simplify operations instead of providing functions for the same? For example normal operations like input and output are good candidates for overloading.

vii. The class functions if defined inside or outside the class must have some justification for doing so.

viii. Where ever applicable, are inline functions used?

ix. Is necessary friend defined? Does the student have a proper justification of why only few functions are friend or why has he made a whole class as friend?

x. Is appropriate and proper inheritance done wherever necessary and applicable?

xi. Is use of containership done where required? Why has he chosen containership where he has applied?

xii. Is generic programming performed for multi – use of the same component?

xiii. Are applicable STL components used or not? If used then does then it should be checked whether the student has a knowledge of other alternative STL components that can be used for the same purpose

xiv. If the solution is better done by using virtual function or RTTI or using casting operators like dynamic_cast, is it really done? In case not, then the student should know the reason of why not applied the same?

xv. Are namespaces well defined?

xvi. Are files used for helping multiple instances of the same program to work without taking multiple inputs? Are proper file handling mechanisms incorporated?

xvii. Are all the outputs well formatted using ios – functions / manipulators?

3. All the necessary declarations in the program should be done with proper naming conventions and must be self – explanatory

4. The program should be well indented

5. Proper comments should be provided for better understanding of the code developed