CSCE 240 – Programming Assignment Five

Due: 11:59pm on Wednesday, November 29th

Program Purpose – Derive three classes from the provided Question class

Create a *TrueFalseQuestion* class as a child of the *Question* class. The class will contain one private bool data member for whether the answer to the question is true or false. You will include a constructor with default arguments for the question and answer, accessor and mutator functions for the private data member, and a destructor and Print function. Read the comments in the supplied *truefalsequestion.h* header file for more detailed requirements. Initial tests for the functionality of the *TrueFalseQuestion* class have been provided in the attached *testtf.cc* source file. You are encouraged to create more rigorous tests.

Create a *ShortAnswerQuestion* class as a child of the *Question* class. The class will contain one private string data member for the correct answer to the question. You will include a constructor with default arguments for the question and answer, accessor and mutator functions for the private data member, and a destructor and Print function. Read the comments in the supplied *shortanswerquestion.h* header file for more detailed requirements. Initial tests for the functionality of the *ShortAnswerQuestion* class have been provided in the attached *testsa.cc* source file. You are encouraged to create more rigorous tests.

Create a *MultipleChoiceQuestion* class as a child of the *Question* class. The class will contain three private data member: an int for the number of answer choices, a pointer to a string to hold the dynamically allocated array of answer choices, and a pointer to a bool to hold the dynamically allocated array of true/false false values denoting whether or not each of the corresponding answer choices is correct. You will include a constructor with default arguments for the data members, a copy constructor, accessor and mutator functions described in the header file, and a destructor and Print function. Read the comments in the supplied *multiplechoicequestion.h* header file for more detailed requirements. Initial tests for the functionality of the *MultipleChoiceQuestion* class have been provided in the attached *testmc1.cc* and *testmc2.cc* source files. You are encouraged to create more rigorous tests.

Additional Specifications

- Do not modify any of the code in *question.h*
- Add all code for the definition of the *TrueFalseQuestion* class to the attached *truefalsequestion.h* header file
- Include all of the necessary code for the *TrueFalseQuestion* class, including the implementation all of the public member functions, in the attached *truefalsequestion.cc* source file.
- Add all code for the definition of the *ShortAnswerQuestion* class to the attached *shortanswerquestion.h* header file
- Include all of the necessary code for the *ShortAnswerQuestion* class, including the implementation all of the public member functions, in the attached *shortanswerquestion.cc source file.*
- Add all code for the definition of the *MultipleChoiceQuestion* class to the attached *multiplechoicequestion.h* header file

- Include all of the necessary code for the *MultipleChoiceQuestion* class, including the implementation all of the public member functions, in the attached *multiplechoicequestion.cc* source file.
- You are required to use pointers as data members in the *MultipleChoiceQuestion* class and to manage the dynamic allocation and deallocation of memory for these data members.
- You will submit a zip file (only a zip file will be accepted) containing *truefalsequestion.h, truefalsequestion.cc, shortanswerquestion.h, shortanswerquestion.cc, multiplechoicequestion.h* and *multiplechoicequestion.cc* to the assignment in Blackboard.
- Source files must compile and run on a computer of the instructor's choosing in the Linux lab (see your course syllabus for additional details).

Initial Testing

- A makefile has been included to aid in using *testtf.cc, testsa.cc, testmc1.cc, testmc2.cc,* and *checkit.cc* to test the basic functionality of your classes.

  To use the *makefile*, your program 5 directory should include:

  your files: *truefalsequestion.h, truefalsequestion.cc, shortanswerquestion.h, shortanswerquestion.cc, multiplechoicequestion.h, multiplechoicequestion.cc*

  the *makefile* and *question.h*

  the test files: *testtf.cc, testsa.cc, testmc1.cc, testmc2.cc,* and *checkit.cc*

  the subdirectory "output" that contains *correcttf.txt, correctsa.txt, correctmc1.txt, correctmc2.txt*

- To test your *TrueFalseQuestion* class, type: *make testtf*
- To test your *ShortAnswerQuestion* class, type: *make testsa*
- To test your *MultipleChoiceQuestion* constructor with default arguments and Print function, type: *make testmc1*
- To test your *MultipleChoiceQuestion* copy constructor and SetAnswerChoices function, type: *make testmc2*
- Note: each of the tests above runs the executable generated by the included *checkit.cc* source file which compares the output created by your functions to the expected output (held in the files provided in the output folder).
- The only header files that may be included in your code are: string, iostream, question.h, truefalsequestion.h, shortanswerquestion.h, and multiplechoicequestion.h. Files that include other headers will not be eligible for correctness points.

Grade Breakdown

Style *truefalsequestion.h*: 0.1 point
Style *truefalsequestion.cc*: 0.2 point
Style *shortanswerquestion.h*: 0.1 point
Style *shortanswerquestion.cc*: 0.2 point
Style *multiplechoicequestion.h*: 0.2 point
Style *multiplechoicequestion.cc*: 0.2 point
Documentation: 1 point
Clean compilation of *truefalsequestion.cc*: 0.2 point

Clean compilation of *shortanswerquestion.cc*: 0.2 point
Clean compilation of *multiplechoicequestion.cc*: 0.2 point
Clean compile/link with *testtf.cc*: 0.1 point
Clean compile/link with *testsa.cc*: 0.1 point
Clean compile/link with *testmc1.cc*: 0.1 point
Clean compile/link with *testmc2.cc*: 0.1 point
Passes instructor's modified *testtf.cc* tests: 2 points
Passes instructor's modified *testsa.cc* tests: 2 points
Passes instructor's modified *testmc1.cc* tests: 1 point
Passes instructor's modified *testmc2.cc* tests: 2 points

The penalty for late assignment submissions is 10% per day up to three days after the assignment due date. No assignment submissions will be accepted more that 3 days after the due date.