

Answer key

0_1_2_3_4_5_6_7_8_9_10_11_12_13_14_15_16_17_18_19_20 How grades were computed. You have a raw score that I computed by counting all questions but one as worth seven points, and that one (question 20) was worth 4 points. Your raw score is out of 100 possible points. I then adjusted the scores in two ways.

First, I was convinced by a student's comments in his test that question 20 (in the test name list, 21 in the list of questions below) was unfair and hinged on a corner case. Therefore, if you missed that question I subtracted 4 points from the number of possible points. Second, exceptions were not on the list of study topics. For each of these questions you missed I subtracted 7 points from the number of possible points. Thus if you missed question 20 and one exception, your score was out of 89 possible points. I then computed a new average. If you got these questions right you received credit for that, and if you didn't get them right you received credit as if they were not on the test.

The list of numbers on the front of your test is the list of questions from the total pool of questions on your test. Thus if the list starts `i_j_k_ ...` questions `i+1`, `j+1` and `k+1` from the file `test2_2012.pdf` were on your test.

Question 1: a. The functions `print` and `print(int)` are virtual, and invoked via a pointer reference. Therefore, the function associated with the object is called regardless of the pointer type, and the D function is called in both cases.

Question 2: b. The functions `print` and `print(int)` are not virtual, and therefore the function called will be the function in the object with the same type as the the pointer used to invoke the function.

Question 3: c. Here the functions `print` and `print(int)` are virtual, but they are not accessed via a pointer or a reference, and so they are treated as if they are not virtual, and the function called is based on the type of the object specified at the call, not the actual type of the object.

Question 4: d. Java always treats all functions as virtual, and therefore it calls the functions associated with the actual object, regardless of how they are called.

Question 5: e. The first call to `f(int)` does not throw an exception, and `cout << ">= 0 "` is executed in `f`. After `f` returns the value of `i` is printed, which is `"0"`. In the second call of `f`, with an argument of `-1`, an exception is thrown and caught, and `"exception caught"` is printed, followed by `"the end."`.

Question 6: a. The call to `f(int)` in the first iteration of the `i` loop. Because `i = 1`, an E exception is thrown, and caught, causing `"e.print()"` to print `"Bad things happened"`. In the next iteration of the loop `i = 0`, and no exception is thrown when `f` is called. Therefore the print after the call to `f(i)` is executed, printing `"0"`. Finally, after the loop terminates `"terminating program"` is printed.

Question 7: b. When `f` is called, `p` is passed by value, i.e. a copy of it is made and passed to `f`. Since the value of the copy is the same, `*p = 58` changes the value of `i` to 58. The copy of `p`, but not `p`, is set to 0. Thus the value of `p` is unchanged..

Question 8: c. When `foo(b)` is called, a copy of `b` is made. The copy of `b` (`bb` refers to the same object as `b`, and therefore the statement `bb.i = 58;` sets the field of the object to 58. `bb` is then set to null, but the value of `b` is unchanged, since `bb` is a copy of `b`.

Question 9: d. You cannot instantiate an abstract class in C++, and `circumference` is an abstract function (because of the `"= 0"`. Thus both `b` and `c` are correct. You should have received 3.5 points for selecting `b` or `c`, but not `d`.

Question 10: e. As with C++, you cannot instantiate abstract classes in Java. Thus `e.` is the answer. Note that abstract classes can declare data members.

Question 11: a. This is a legal program, but Test must define all methods in all interfaces that it implements.

Question 12: b. It is legal for Shape and Color to both declare and define an i member, but if some class implements both it cannot legally refer to i.

Question 13: c. A list allows $O(1)$ inserts and deletes from the front, back and middle. The others will have worst complexity for at least one of these.

Question 14: d. (a) is not right because any function in any class, or any global function, can legally accept MyComplex arguments. (b) is wrong because the implied (this) argument of the function if implemented in MyComplex should be of type MyComplex, but it has to be a stream. (c) is wrong because there is no such requirement, and (d) is right.

Question 15: e. The member function operator- is a binary operator, and the global (non-member) function is a unary operator. So (a) and (c) are wrong and (b) is right. (d) is also right, so (e) is the best answer. If you answered (b) or (d), but not (e) you should have received 3.5 points.

Question 16: a. call 1 only requires a conversion of float to double to match foo 1, and this is a closer match than any other, so this is what is done. call 2 requires a conversion of f to a double and L to an int, which is closer than another other target function, and so is a match. call 3 can either convert the first (double) argument to an int and call call foo 1, or convert the second (double) argument to an int and call foo2, but these are equally difficult (same specificity level, same number of conversions) and so it is ambiguous which should be called. call 4 is an exact match of foo 3. Thus a is the right answer.

Question 17: b. call 1 calls foo 1 since only a float to double widening conversion is required, and no other call matches as closely. call 2 has no match as it requires converting a long to an int, which can lose information and is not done by default by Java. call 3 is an exact match for foo 3. Thus b is the correct answer.

Question 18: c. The D2 constructor initializes memory for the D2 object, and (implicitly) calls the zero-arg D1 constructor, which initializes its memory and (implicitly) calls the zero-arg B constructor which it initializes its memory. The body of the B constructor is then executed, which prints B, returns to the D1 constructor whose body is executed, printing D1. It then finishes and returns to the D2 constructor, which executes its body, printing D2. Thus c is correct.

Question 19: d. See the answer to 18.

Question 20: e. Initializations specified in a constructor's initializer are performed in the order the symbols being initialized were declared.

Question 21: a. The private qualifier says that a field/function is private to a class, thus any object of the class can access the private fields of any other object instantiated from the class.