1: Square:

2: Shape ---- Default Constructor

3: Square -- Default Constructor

4: Square -- Destructor

5: Shape -- Destructor

6: Shape -- Constructor

7: Square -- Constructor

8: Square -- Destructor

9: Shape -- Destructor

10:

11: Circle:

12: Shape ---- Default Constructor

13: Circle -- Default Constructor

14: Shape -- Constructor

15: Circle -- Constructor

16: Shape -- Copy Constructor

17: Circle -- Copy Constructor

18: Circle -- Assignment Operator

19: Circle -- Destructor

20: Shape -- Destructor

21: Circle -- Destructor

22: Shape -- Destructor

23: Circle -- Destructor

24: Shape – Destructor

Line 1: Comes from the print stamen on line 49 in the main

Line 2-3: These lines are printed because of initialization of “square\_ptr” on line 50 with the Default Constructor of Square using new. The default constructor of Square First calls the default constructor of the base class Shape before running its own constructor so Shape is printed before Square

Lines 4-5: These are printed because of the delete statement of “square\_ptr” on line 50 deallocating the memory. The derived class’s destructor is called before the destructor of the base class so the Square statement is printed before Shape

Lines 6-7: Theses are printed because of the initialization of “square\_var” as a variable using the regular constructor inside brackets on line 52. The base class’s constructor is called before the derived classes constructor so Shape is before Square

Lines 8-9: These lines are printed because the program moved out of the brackets that “square\_var” was initialized in (line 52) and removing the data from the stack that it was stored in calling the destructor. The derived class’s destructor is called before the destructor of the base class so the Square statement is printed before Shape

Lines 10-11: Comes from the print statement on line 54

Lines 12-13: These lines are printed because of initialization of “circle\_ptr” on line 55 with the Default Constructor of Circle using new. The default constructor of Circle First calls the default constructor of the base class Shape before running its own constructor so Shape is printed before Circle

Lines 14-15: Theses are printed because of the initialization of “circle\_var1” as a variable using the regular constructor on line 56. The base class’s constructor is called before the derived classes constructor so Shape is before Circle

Lines 16-17: Theses are printed because of the initialization of “circle\_var2” as a variable using the copy constructor on line 57. The base class’s copy constructor is called before the derived classes copy constructor so Shape is before Circle

Line 18: This line is printed because of the assignment operation on line 58 where circle\_var1 is set to the value of the dereferenced “circle\_ptr”. The assignment calls the assignment operator

Lines 19-20: These are printed because of the delete statement of “circle\_ptr” on line 59 deallocating the memory. The derived class’s destructor is called before the destructor of the base class so the Circle statement is printed before Shape

Lines 21-24: These are printed because the execution reached the end of the main function and returned 0 removing variables in the stack so the destructors of circle\_var1 and circle\_var2 are called. The derived class’s destructor is called before the destructor of the base class so the Circle statement is printed before Shape in each case