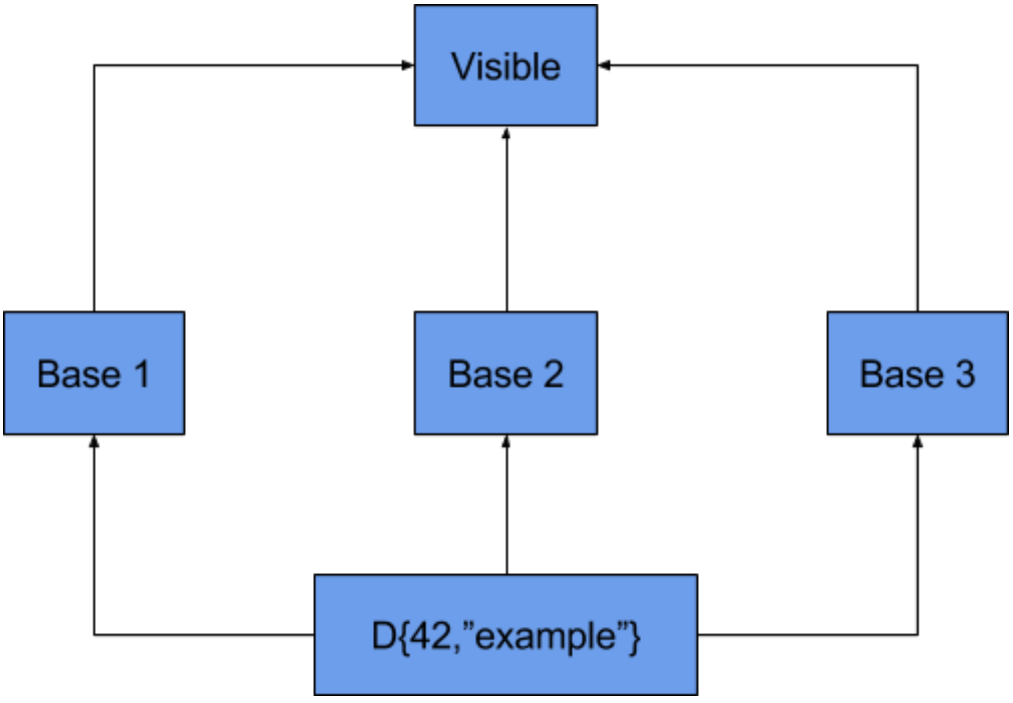
UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
INSTITUTO METRÓPOLE DIGITAL
IMD0030 - Linguagem de Programação I - T03

Discentes: Cleydson Talles Araújo
Fernando Rodrigues Maciel
Gabriel Queiroz de Almeida Pereira

PROBLEMA 7 -

Q1:



Q2:

a) É justamente o que é impresso na tela. Ele chama o construtor *Animal* duas vezes, já que ela é a classe mãe e logo em seguida chama o construtor das duas derivadas, que é *Mammal* e *WingedAnimal*. E por fim chama a classe *Bat*, que não promove nada.
b) O erro do compilador é resultado da não especificação de qual classe o objeto *"&mammal"* devia seguir. Se era *Mammal* ou *WingedAnimal*.
c) O esperado. A mesma execução
d) Como a classe agora é virtual, não se é preciso chamar o construí-la novamente. Com isso, a classe animal só é chamada agora uma vez.
e) Não podemos pois a classe mammal não tem e nem herda o método flap.

Q3:

a) Uma função declarada *friend* em uma classe, permite acesso aos membros privados e protegidos desta classe,apenas essa função receberá permissão de acesso aos membros privados que faz a declaração. Já uma classe declarada friend,

permite que a outra classe tenha acesso aos dados delas, sem ser restritamente através de uma função.

b) Seu uso típico é quando você quer ter acesso aos atributos de uma classe, e estes são privados ou protegidos.
c) Ele mostra que os atributos “width” e “height” são privados e não podem ter acesso a esses dados.

Q4:

a) Ele serve para dizer que tem uma classe Square, já que na linha 13 ele utiliza essa classe como parâmetro. Aí logo em seguida ele declara a classe Square detalhadamente. Funciona na mesma forma que funções void. Tem-se que declarar ela antes da função que seja utilizada.
b) A relação é que na linha 13 ele chama a função Square, que já havia sido previamente declarada. Caso seja apagado a linha 6, dará error.
c)

Q5:

a) A função “assert” funciona para o compilador verificar se os dois parâmetros passado são iguais. Caso não seja, ele gera uma mensagem de erro depois que o programa é executável.
b) Ele diz que a função assert falhou e mostra que os dois parâmetros comparados não são iguais, logo em seguida aborta o programa.

Q6:

a) O programa se encontra no diretório lists/list0901v2.cpp
b) O programa se encontra no diretório lists/list0901v3.cpp

Q7:

a) OK.
b) O programa se encontra no diretório lists/list1004.cpp
c) O programa se encontra no diretório lists/list1005.cpp

Q8:

a) Ele lê o inteiro “x” até ser diferente do tipo int e a cada vez que lê armazena no final do vetor Data. Quando interrompido, ele entra no laço for. O laço *for* cria um iterador start, que é o primeiro elemento do vetor. E o iterador end, que é o último espaço vazio do vetor. Assim que inicia o FOR end é diminuído em um, para ficar no último espaço ocupado e há um “swap” entre os iteradores, o primeiro vai pro último e vice-versa. Logo em seguida, o start é diminuído em um e o end também. É feito isso até que o vetor esteja totalmente invertido.

Q9:

O programa se encontra no diretório lists/list1501v2.cpp

Q10:

a) O programa se encontra no diretório lists/list1501v3.cpp

Q11:

a) O programa se encontra no diretório lists/list1504.cpp

Q12:

O contêiner utilizado foi a stack para implementar uma calculadora polonesa com as quatro operações básicas.