Call 1:
This call is located on Line 74. It has a static and dynamic type of B with a signature of h(). Since function h() does not exist in class B, program tries to find one in the parent class, specifically on Line 6, which just prints A.h().

Call 2:

On Line 76 on a D type object function with signature h() is called. Since object is static type and dynamic type of D, program tries to find appropriate function in class D. There are two compatible ones, but function on Line 39 does not need an input and one on Line 43 does. First one is going to be chosen since no input is given. First this function would call h() from a super class and then print D.h(). According to same logic h() function with no inputs is going to be chosen from C class which calls h() function from its parents and then prints C.h(). Since there is no h() function in class B then it is going to search h() function in class A, which just prints A.h(). Finally as an output we are going to have:

A.h()

C.h()

D.h()

Call 3:

On Line 78 on an object "ac" which has a static type of A is changed dynamically multiple times to type C. Then function h() is called on it. There are two compatible methods in class C(Lines 26, 30), but first one is going to be executed since no input is given. Then it is going to call for h() function from its super class and print C.h(). Since no function with a signature h() is in class B, one in the class A is going to be executed which prints A.h().Finally as an output we are going to have:

A.h()

C.h()

Call 4:

Line 80. On an object "x" which is of X type, function of signature f() is called with input d, which is casted to B type. There are two function of f() in class X. One on Line 52 requires input of type S(sub of B) and second one on Line 56 which requires D type variable as an input. Logically first one is going to be chosen. This function prints X.f(S) and calls a function f() on type S object with input x. There is a function on Line 16 which satisfies these properties. It prints B.f(X) and calls function g() on X type with input null. On Line 49 there is g() function which is called and it prints X.g(C). Finally as an output we are going to have:

X.f(S)

B.f(X)

X.g(C)

Call 5:

Line 82. On X type objet function with signature f() is called with a newly created object C. Since we are creating a new C object, a constructor for that object and its parents are going to be called. C's parents are B and A. Since constructor just prints its class type and parenthesis. First it is going to print A(), B() and C(). Then in class X there are two f() functions. First (Line 52) takes S type object as an input and Second (Line 56) takes D type object as an input. First one is going to be chosen since S and C have same parent B. X.f(S) is going to be printed and then same logic follows as in Call 4. Finally as an output we are going to have:

A()

B()

C()

X.f(S)

B.f(X)

X.g(C)

Call 6:

Line 84. On X static type object function with signature f() with input of type D is called. As you know in class X there are two f() functions, but in this case second one is going to be chosen, because we have D type object as an input. It is going to print X.f(D a) and call function h() on a static D type object a with input of casted C type a. In class D there are two h() function one requires no input and second one on Line 43 takes input type C variable. This function calls h() from its parent class with casted B type variable. In class C there are two h() function one requires input other does not. Thankfully the function on Line 30 satisfies requirements and so logically is going to be executed. Now we search for function h() in the and appropriately there is one in class D on Line 39. I have already explained how it works in the Call 2. Finally as an output we are going to have:

X.f(D a)

A.h()

C.h()

D.h()