1 )The difference between Constructor and method are

a. The number of arguments
b. The return type(with and without) // Constructors have no return type
c. The naming constraint

1) a&c
2) a&b
3) b&c
4) a&b&c

**Answer: (3) B&C**

2) In the below, which lines will have error?

a) Public abstract class AnAbstract
b) Private abstract void doB(int x,double y);}
c)
d) Private void doA(){...}

1) Line 2
2) Line 3
3) Line 2 & line 3
4) None of the lines
5) All of the lines
6) Line 1 & 2
7) Line 1
8) Line 1 & line 3

**Answer: (2) Line 3 - Abstract shouldn't have any input or implementation.**

3) Square s1, s2;
s1 = new Square();
s2=s1;
s2.setWidth(6);
s1.setWidth(99);
System.out.println(s2.getWidth());

a) 6
b) 100
c) 99
d) None of the options
e) 105

**Answer: (C) 99**

4) Student class is a subclass of Person class. What is the output of the below code:

```
Person p = new Student();
If (p instanceof Object)
        System.out.println("Object instance");
else if (p instance of Person)
        System.out.println("Person instance");
else
        System.out.println("Other instance");
```

- Other instance
- Object instance
- Error
- Person instance

**Answer: (B)- Object instance**

5) As long as the same message is sent to any objects, polymorphism behaviour is always possible.

- False
- True

**Answer:False**

polymorphic. In Java, all Java objects are polymorphic since any object will pass the IS-A test for their own type and for the class Object.

We know thatonly possible way to access an object is through a reference variable. A reference variable can be of only one type. Once declared, the type of a reference variable cannot be changed.

6) A software Object's behavior is exposed through___

- requests
- messages
- states
- methods
- all of the options
- none of the options

**Answer: (D)- Methods**

7) The below codes correctly show Method Overloading

```
Public class Dog{
        public int move(){.....}
        public void move(int x, double y){....}
        public void move(){....} //same parameter input tats y false
}
```

a) False
b) True

**Answer:(A)- False**

Same Method Name but:
1) different number of parameters.
2) different parameter *types*.

OR

8) With Inheritance relationship, method overriding will always exist.
- True
- False

**Answer: False**

9)When all methods are implemented in a class, this class is known as
- Concrete class
- Interface
- Working class
- Abstract class
- None of the options
- Subclass
-

Note: A Concrete class has implementation for all methods, i.e. NO abstract methods.

**Answer: (A)- Concrete Class**

10) A Software object's ___ is stored in fields/attributes
a) Name
b) None of the options
c) Behaviours
d) Methods
e) States
f) All of the options

**Answer: (E)- States**

State/ Properties: Color, maximum speed, etc.

Behavior/ Methods: Turn, brake and accelerate

11)Given a person class, doing the below
Person p =(Person)new String(“Tom”);
Will give compilation error due to absence of inheritance relationship between Person and String

How about the below code?
Object o  =new String(“Tom”);//line 1
Person p1=(person)o; //line2 < Downcasting, that's why error

a) No Compilation Error
b) Compilation Error in line 1
c) Compilation Error in line 2

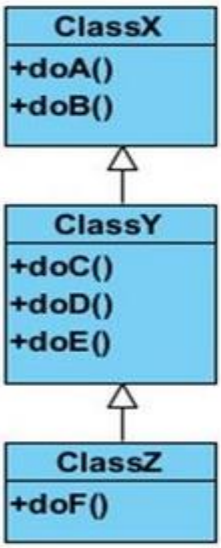
**Answer: (C)**

12) Object interact with objects via___

- Signalling
- Talking
- Messaging
- None of the options
- Data passing

**Answer:Messaging**

13)



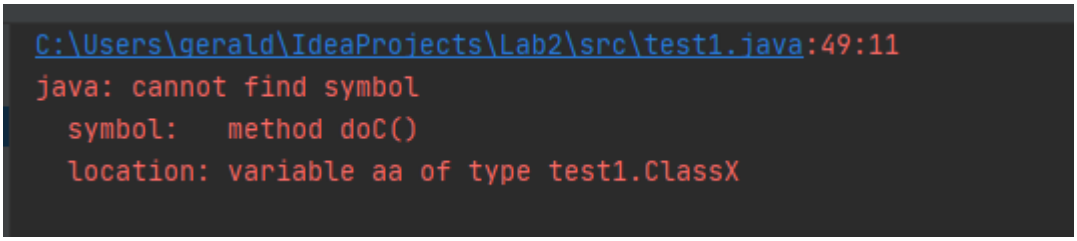

Given the above class hierarchy, which message/statement (a-d) will have compilation error:

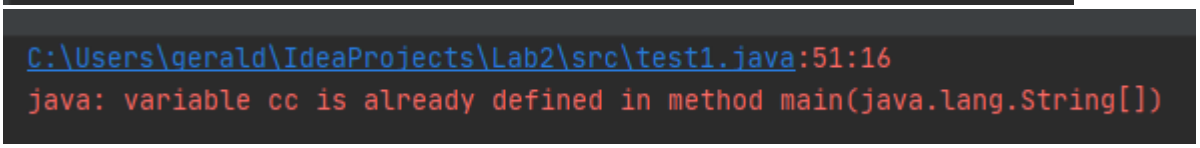
ClassX aa= new ClassY(); //upcast ok
ClassZ bb = (ClassZ)aa // aa(ClassY) not instance of ClassZ(rt error) ClassY cannot be cast to class ClassZ(FAIL)
ClassY cc=bb; // same as ClassY cc = (ClassZ)aa

```
Exception in thread "main" java.lang.ClassCastException Create breakpoint : class test1$ClassY cannot be cast to class test1$ClassZ (test1$ClassY and test1$ClassZ are in unnamed module of loader 'app')
    at test1.main(test1.java:46)
```

```
bb.doA();
aa.doC();
cc.doB();
ClassY cc=bb;
```

```
public static void main(String[] args) {
    ClassX aa = new ClassY();
    ClassZ bb = (ClassZ)aa;
    ClassY cc=bb;
    bb.doA();
    aa.doC();
    cc.doB();
    ClassY cc=bb;
}
```




a)bb.doA() //
b)aa.doC() //doC not found in X
c)cc.doB()
d)ClassY cc=bb;

(A)- None of the options
(B)- a
(C)- c
(D)- d
(E)- b
**Answer:E -** aa.doC()

14) With implements(interfaces) relationship, method overriding will always exist.

- True
- False

**Answer: True**