1. Object oriented programming is a programming style that mimics real world scenarios. We use OOP to solve real world problems. In OOP all the related functions and variables are bundled together under a single entity.
OOP offers quite a few features such as Inheritance, Abstraction, Encapsulation and polymorphism

2. Created all the programs and they are placed in the src directory.

MCQ

1. A
Technically this is a wrong question since there is no virtual keyword in java

2. A
Instance of the interface cannot be created other than this all other options are correct.

3. B
Overloading is also known as compile time polymorphism.

4. A
default constructor does not have any parameter

5. A
In java there is no concept of pointers hence there is no Arrow operator. For c++ the answer will be C

6. D

7. D
A non-member function cannot access anything without creating an instance of the class

8. 0
The variable i will be assigned its default value
9. A

10. Derived::show() called
Since the method is overridden in the derived class. This is a classic example of runtime polymorphism.

11. Error
we cannot override a final method in the derived class

12 "Base::show() called"
The method to be called will be decided at compile time since static methods cannot be overridden. The method to be called will be decided by the type of reference variable

13. Test class Derived class
The test class instance calls its own getDetails method which calls the superclass(Derived) class getDetails method

14. Compile time error
Since we are trying to override the getDetails() methods but the return type int cannot be used since it is incompatible with void

15. Adding to 100, x = 104
Adding to 0, y = 3 3 3
x and y are static variables so they belong to the class and shared by all the instances

16. error: reference to m1 is ambiguous
We are trying to accomplish method overloading, and the best match method will be called, However in this case the best possible match is indeterminate

17. error: incompatible types:
We cannot assign a null value to primitive dataType variable

18. 0 0
Assuming Main and Test classes are in the same package therefore we can even access the protected Variable in a class that is not inherited. The default values will be assigned to the int variables which is 0

19. Constructor called 10
Constructor called 5
Variable t1 is one of the members of t2 which gets created before the constructor is called for t2 and in the constructor of t2 we are again
reassigning t1 with a new Test1 instance. In short we are creating the Test1 instance twice therefore the Test1's constructor will be called twice

20. 7
Although the code int []x[] is not very readable but the compiler will treat x as the array name for the 2D array

21. 2
The method that will be called will be decided at runtime, This is a classic example of runtime polymorphism

22. 2
This question is pretty similar to question 21

23. 1 2
In this case basically we are hiding the variable j inherited from class A
24. 1 2
We are inheriting the data members from the parent class which gets initialised through the call super()

25. obj1.a = 4 obj1.b = 3
obj2.a = 4 obj1.b = 3
The only tricky part in the question is the concept of prefix and postfix operators. They look quite similar but behave quite differently