10. abstract public class Employee {

11. protected abstract double getSalesAmount();

12. public double getCommision() {

13. return getSalesAmount() * 0.15;

14. }

15. }

16. class Sales extends Employee {

17. [protected double getSalesAmount() { return 1230.45; } ⬍] ✔

18. }

Which method, inserted at line 17, correctly complete the Sales class?

Predict the output of the following program:

```
abstract class Demo
{
    public int a;
    Demo()
    {
        a = 10;
    }
    abstract public void set();
    abstract final public void get();
}
class Test extends Demo
{
    public void set(int a)
    {
        this.a = a;
    }
    final public void get()
    {
        System.out.println("a = " + a);
    }
    public static void main(String[] args)
    {
        Test obj = new Test();
        obj.set(20);
        obj.get();
    }
}
```

Select one:

○ a = 10

◉ Compilation error ✔

○ a = 20

Will the following code get executed successfully ?

abstract class Shape

{

   int i = 111, j = 222;

   abstract void calcArea();

   abstract void calcVolume();

}

abstract class Square extends Shape

{

   void calcVolume() { System.out.println(j); }

   void calcArea(){ System.out.println(j); }

}

public class Test

{

   public static void main(String[] args)

   {

     Square c = new Square();

      c.calcArea();

      c.calcVolume();

   }

}

Select one:

○

   Compilation error. The class Square must not be declared abstract as it has no abstract methods.

○ The code will get executed successfully.

◉ Compilation error. Object cannot be created for the class Square as "Square c = new Square() ". ✔

Will the below code will execute successfully ?

abstract class Shape

{

   final abstract int calcArea();

}

Select one:

○ True

◉ False ✔

What is the output of the given code?

```
abstract class Shape
{
   int i = 111, j = 222;
   abstract void calcArea();
   abstract void calcVolume();
}
 abstract class Quadrilateral extends Shape
{
   void calcArea()
   {
      System.out.println(i);
   }
}
class Square extends Quadrilateral
{
   void calcVolume()
   {
      System.out.println(j);
   }
}
public class Test
{
   public static void main(String[] args)
   {
      Square c = new Square();
       c.calcArea();
       c.calcVolume();

   }
}
```

Select one:

○ Compile time error because 'class Square' is not override all the abstract methods, so should declare it as 'abstract'

○ Compile time error because trying to instantiate the 'class Square' which does not override all the abstract methods

◉ 111
222 ✔

○ Run time Error

Question 6

Correct

Mark 1.00 out of 1.00

⚑ Flag question

What is the Output of following Java Program?

```java
abstract class Demo
{
    public int a;
    Demo()
    {
        a = 10;
    }
     abstract public void set();
}
class Test extends Demo
{
    final public void get()
    {
        System.out.println("a = " + a);
    }
     public static void main(String[] args)
    {
        Test obj = new Test();
        obj.get();
    }
}
```

Select one:
- ○ Runtime Exception
- ○ a=10
- ● Compile Time Error ✔

Question 7

Correct

Mark 1.00 out of 1.00

⚑ Flag question

An abstract class [can ⇕] ✔ have non abstract methods also.

Question 8

Correct

Mark 1.00 out of 1.00

⚑ Flag question

Abstract methods [cannot ⇕] ✔ be final.