Started on	Wednesday, 1 April 2020, 5:24 PM
State	Finished
Completed on	Wednesday, 1 April 2020, 5:29 PM
Time taken	4 mins 19 secs
Marks	7.00/8.00
Grade	**87.50** out of 100.00
Feedback	Congratulations!! You have passed by securing more than 80%

Question 1

Correct

Mark 1.00 out of 1.00

An abstract class can ✔ have non abstract methods also.

Your answer is correct.

The correct answer is:
An abstract class [can] have non abstract methods also.

Question 2

Correct

Mark 1.00 out of 1.00

```
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?

Since the Sales class is not abstract, it must have the implementation for the abstract method in Employee class. We must not reduce the visibility of the method in the child class. Hence, protected double getSalesAmount() { }

The correct answer is:

```
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?

Correct

Mark 1.00 out of 1.00

Your answer is correct.

The correct answer is:
Abstract methods [cannot] be final.

Correct

Mark 1.00 out of 1.00

```
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:

- a. Compile time error because trying to instantiate the 'class Square' which does not override all the abstract methods
- b. Compile time error because 'class Square' is not override all the abstract methods, so should declare it as 'abstract'
- c. Run time Error
- d. 111
  222 ✔

Here is multilevel inheritance. The main method has object created for class Square, referred by Square type reference.

The abstract methods clacArea() and calcVolume() in parent have their implementation within their children. Only when an implementation is not found child class, it is searched for in parent class and executed. Hence 111, 222.

The correct answer is: 111
222

Correct

Mark 1.00 out of 1.00

```
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. a = 20
- b. a = 10
- c. Compilation error ✔

Your answer is correct.

The correct answer is: Compilation error

Question 6

Incorrect

Mark 0.00 out of 1.00

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:

- a. Yes, the code will get executed successfully. ✖
- b. No – Compilation error.

Object cannot be created for abstract class.

The correct answer is:
No – Compilation error.

Correct

Mark 1.00 out of 1.00

```
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:

- a. Compile Time Error ✔
- b. Runtime Exception
- c. a=10

The program produces compile time error because the abstract set method in the parent has not found its implementation in child class (which has the main method too).

The correct answer is: Compile Time Error

Question 8

Correct

Mark 1.00 out of 1.00

Will the below code will execute successfully ?

```
abstract class Shape
{
   final abstract int calcArea();
}
```

Select one:

- True
- False ✔

Since the abstract method within the abstract class is declared final, it can never be implemented in any of its child classes. Hence, it is an error.

The correct answer is 'False'.