Started on	Friday, 27 March 2020, 1:27 AM
State	Finished
Completed on	Friday, 27 March 2020, 1:33 AM
Time taken	6 mins 11 secs
Marks	16.00/16.00
Grade	**100.00** out of 100.00
Feedback	Congratulations!! You have passed by securing more than 80%

Question 1

Correct

Mark 1.00 out of 1.00

Assume Book is a parent class and Magazine class is the child of Book class.

Match the following:

Book b = new Magazine();	Upcasting ✔
Magazine m = (Magazine) b	Downcasting ✔

Your answer is correct.

The correct answer is: Book b = new Magazine(); → Upcasting, Magazine m = (Magazine) b → Downcasting

Correct

Mark 1.00 out of 1.00

1. public class Employee {
2. String name;
3. double baseSalary;
4. Employee(String name, double baseSalary) {
5. this.name = name;
6. this.baseSalary = baseSalary;
7. }
8. }

And:

11. public class Salesperson extends Employee {
12. double commission;
13. public Salesperson(String name, double baseSalary,
14. double commission) {
15. // insert code here
16. }
17. }

Which code, inserted at line 17, completes the Salesperson constructor?

Select one:

- a. super(name, baseSalary);
  this.commission = commission; ✔
- b. super();
  this.commission =commission;
- c. super();
  commission = commission;
- d. this.commission = commission;
  super();
- e. this.commission = commission;
  super(name, baseSalary);
- f. this.commission = commission;

Your answer is correct.

The correct answer is: super(name, baseSalary);
this.commission = commission;

Question 3

Correct

Mark 1.00 out of 1.00

A default method in an interface can be either private or public or protected. State True or False.

Select one:

- True
- False ✔

The correct answer is 'False'.

Correct

Mark 1.00 out of 1.00

Select one:

a.
public abstract class Shape { public Square draw(); }

b.
public class Shape { public abstract Square draw(); }

c.
public abstract class Shape { public Square draw() { } } ✔

d.
public class Shape abstract { public abstract Square draw(); }

Your answer is correct.

The correct answer is:
public abstract class Shape { public Square draw() { } }

Question 5

Correct

Mark 1.00 out of 1.00

```
public abstract class Shape  {
    private int x; private int y;
    public abstract void draw();
    public void setAnchor(int x, int y) {
            this.x = x;
            this.y = y;
    }
}
```

Which two classes use the Shape class correctly? (Choose two.)

Select one or more:

a.
```
public class Circle extends Shape  {
    private int radius;
    public void draw();
}
```

b.
```
public abstract class Circle extends Shape {
     private int radius;
}
```
✔

c.
```
public class Circle extends Shape   {
    private int radius;
    public void draw() {/* code here */}
}
```
✔

d.
```
public class Circle implements Shape  {
    private int radius;
}
```

Your answer is correct.
If a class inherits an abstract class with abstract methods, it should provide implementation for all the abstract methods in the parent. If not, then that class needs to be declared as abstract.

The correct answers are: public abstract class Circle extends Shape {
    private int radius;
}, public class Circle extends Shape   {
    private int radius;
    public void draw() {/* code here */}
}

Correct

Mark 1.00 out of 1.00

Select one:

- a. final ✔
- b. protected
- c. public
- d. private

Your answer is correct.

The correct answer is: final

Question 7

Correct

Mark 1.00 out of 1.00

If a method in a super class is overridden by the sub class, then the overridden method can be invoked using _____ keyword.

Select one:

- a. class
- b. super ✔
- c. this
- d. extends

Your answer is correct.

The correct answer is: super

Question 8

Correct

Mark 1.00 out of 1.00

If a class inheriting an abstract class does not provide definition for all abstract methods in the parent class, then it will be known as _____.

Select one:

- a. Static class
- b. A concrete class
- c. abstract ✔
- d. A simple class

Your answer is correct.

The correct answer is: abstract

Question 9

Correct

Mark 1.00 out of 1.00

_____ can be achieved through inheritance.

Select one:

- a. code reusability
- b. run time polymorphism
- c. both run time polymorphism & code reusability ✔
- d. none of the options

Your answer is correct.

The correct answer is: both run time polymorphism & code reusability

Correct

Mark 1.00 out of 1.00

Select one:

- a. static, default ✔
- b. public , abstra
- c. abstract, default
- d. abstract, static

Your answer is correct.

The correct answer is: static, default

Question 11

Correct

Mark 1.00 out of 1.00

Which of the following represents the correct lambda expression for the Functional method :

int findMax(int a,int b) ?

Select one:

- a.
```
(a, b) -> {
 int min = a>b ? a : b;
   }
```
- b.
```
(int a,int b) ->
 int min = a>b ? a : b;
 return min;
```
- c.
```
(int a,int b) -> {
 int min = a>b ? a : b;
 return min;  }
```
✔
- d.
```
(int a,int b) -> {
 min = a>b ? a : b;
 return min;  }
```

Your answer is correct.

The correct answer is:

```
(int a,int b) -> {
 int min = a>b ? a : b;
 return min;  }
```

Correct

Mark 4.00 out of 4.00

```
class A  {
   String name="A";
   public String getName()  {
      return name;
   }
   String greeting()  {
      return "class A";
   }
}
class B extends A  {
   String name="B";
   String greeting()  {
      return "class B";
   }
}
public class Test  {
   public static void main(String arg[])  {
      A a=new A();
      A b=new B();
      System.out.println(a.greeting()+" has name "+a.getName());
      System.out.println(b.greeting()+" has name "+b.getName());
   }
}
```

Place the names "A" and "B" in the following output.

class [A] ✔ has name [A] ✔

class [B] ✔ has name [A] ✔

Correct

Mark 1.00 out of 1.00

```
public abstract class Abs {
        public Abs(){
                System.out.println("Constructor from Abstract class");
        }
 }
public class Test extends Abs {
        public static void main(String args[]){
                        Abs obj=new Test();
        }
}
```

Select one:

- a. Compile time error: An abstract class cannot have a constructor
- b. Compile time error: An abstract class cannot be instantiated
- c. Constructor from Abstract class ✔
- d. Program will execute successfully but not display anything

Your answer is correct.
In the constructor of the child class, the first line should be a call to the super class constructor. If not written, then implicitly it invokes the super class constructor as super();

The correct answer is: Constructor from Abstract class