



Started on	Saturday, 21 March 2020, 8:33 PM
State	Finished
Completed on	Saturday, 21 March 2020, 8:44 PM
Time taken	10 mins 13 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

Question 2

Correct

Mark 1.00 out of 1.00

If a method in an interface is implemented, then that method should be either _____ or _____.

Select one:

- ☒ a. static, default ✓
- ☐ b. abstract, default
- ☐ c. abstract, static
- ☐ d. public , abstra

Your answer is correct.

The correct answer is: static, default

Question 3

Correct

Mark 1.00 out of 1.00

Which would declare a compilable abstract class?

Select one:

- ☒ a.
public abstract class Shape { public Square draw() {} } ✓
- ☐ b.
public abstract class Shape { public Square draw(); }
- ☐ c.
public class Shape { public abstract 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() {} }



Correct
Mark 1.00 out of 1.00

Select one:

- ☐ a. public
- ☐ b. protected
- ☒ c. final ✓
- ☐ d. private

Your answer is correct.

The correct answer is: final

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 class Circle implements Shape {
 private int radius;
}
- ☒ c. public abstract class Circle extends Shape {
 private int radius;
} ✓
- ☒ d. public class Circle extends Shape {
 private int radius;
 public void draw() { /* code here */ }
} ✓

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

```
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 be instantiated
- ☒ b. Constructor from Abstract class ✓
- ☐ c. Compile time error: An abstract class cannot have a constructor
- ☐ 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



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 ✓ has name ✓
class ✓ has name ✓

Question 8

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

keyword.

Select one:

- ☐ a. class
- ☒ b. super ✓
- ☐ c. extends
- ☐ d. this

Your answer is correct.

The correct answer is: super

Question 10

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

Question 11

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

```
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. this.commission = commission;
super();
- ☐ b. this.commission = commission;
- ☐ c. super();
commission = commission;
- ☐ d. this.commission = commission;
super(name, baseSalary);
- ☒ e. super(name, baseSalary);
this.commission = commission; ✓
- ☐ f. super();
this.commission = commission;

Your answer is correct.

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

Question 13

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. abstract ✓
- ☐ b. A concrete class
- ☐ c. Static class
- ☐ d. A simple class

Your answer is correct.

The correct answer is: abstract