Java Class Design - 0

Consider the code of Test.java file:

```
class Printer {
    private static int count = 0;
    private Printer() {
        count++;
    }

    static Printer getInstance() {
        return PrinterCreator.printer;
    }

    static class PrinterCreator {
        static Printer printer = new Printer();
    }

    static int getCount() {
        return count;
    }
}

public class Test {
    public static void main(String[] args) {
        Printer p1 = Printer.getInstance();
        Printer p2 = Printer.getInstance();
        Printer p3 = Printer.getInstance();
        System.out.println(Printer.getCount());
    }
}
```

What will be the result of compiling and executing Test class?

A - 0

B - 3

C - 1

D - 2

Java Class Design - 1

Consider the code of Test.java file:

```
class Player {
    String name;
    int age;

    Player() {
        this.name = "Yuvraj";
        this.age = 36;
    }

    public String toString() {
        return name + ", " + age;
    }

    public Class getClass() {
        return super.getClass();
    }
}

public class Test {
    public static void main(String[] args) {
        System.out.println(new Player());
    }
}
```

What will be the result of compiling and executing Test class

A - Text containing @ symbol

B - null, 0

C - Compilation error

D - Yuvraj, 36

Java Class Design - 2

Given:

```
public class Initializer {
    static int a = 10000;

    static {
        --a;
    }

    {
        ++a;
    }

    public static void main(String[] args) {
        System.out.println(a);
    }
}
```

What is the result of compiling and executing Initializer class?

A - java.lang.ExceptionininitailizerError

B - 9999

C - 10000

D - Compilation Error

Java Class Design - 3

Consider the code of Test.java file:

```
//Test.java
package com.training.ocp;

class Player {
    String name;
    int age;
    Player() {
        this.name = "Virat";
        this.age = 29;
    }

    public int hashCode() {
        return 100;
    }
}

public class Test {
    public static void main(String[] args) {
        System.out.println(new Player());
    }
}
```

Hexadecimal representation of 100 is 64.

Which of the following option is correct?

A - Code compiles successfully and on execution always prints “com.training.ocp.Player@64” on to the console

B - Code compiles successfully but throws an exception on executing it

C - Code doesn’t compile successfully

D - None of the other options

Java Class Design - 4

Consider the code of Greet.java file:

```
public final class Greet {
    private String msg;
    public Greet(String msg) {
        this.msg = msg;
    }

    public String getMsg() {
        return msg;
    }

    public void setMsg(String msg) {
        this.msg = msg;
    }
}
```

Is Greet class an immutable class?

A - No

B - Yes

Java Class Design - 5

Consider below code:

```
//Child.java
class Parent {
    public void m() {
        System.out.println("Parent");
    }
}

public abstract class Child extends Parent { //Line 9
    public static void main(String [] args) { //Line 10
        new Parent().m(); //Line 11
    }
}
```

What will be the result of compiling and executing Child class?

A - Compilation error at Line 11

B - Compilation error at Line 10

C - Parent

D - Compilation error at Line 9

Java Class Design - 6

Consider the code of Test.java file:

```
class Player {
    String name;
    int age;

    Player(String name, int age) {
        this.name = name;
        this.age = age;
    }

    public String toString() {
        return name + ", " + age;
    }

    public boolean equals(Player player) {
        if(player != null && this.name.equals(player.name)
                && this.age == player.age) {
            return true;
        }
        return false;
    }
}

public class Test {
    public static void main(String[] args) {
        Object p1 = new Player("Sachin", 44);
        Object p2 = new Player("Sachin", 44);
        System.out.println(p1.equals(p2));
    }
}
```

What will be the result of compiling and executing Test class?

A - false

B - Compilation error in Player class

C - true

D - Compilation error in Test class

Java Class Design - 7

Consider the code of Test.java file:

```
class Point {
    private int x;
    private int y;

    Point(){
        Point(10, 20);
    }

    Point(int x, int y) {
        this.x = x;
        this.y = y;
    }

    @Override
    public String toString() {
        return "Point{" + x + ", " + y + "}";
    }
}

public class Test {
    public static void main(String[] args) {
        Point p = new Point();
        System.out.println(p);
    }
}
```

What will be the result of compiling and executing Test class?

A - Compilation error in Test class

B - Point{0, 0}

C - Point{10, 20}

D - Compilation error in Point class

Java Class Design - 8

Consider the code of Test.java file:

```
abstract class Animal {
    public static void vaccinate() {
        System.out.println("Vaccinating...");
    }

    abstract void treating();
}

public class Test {
    public static void main(String[] args) {
        Animal.vaccinate();
    }
}
```

What will be the result of compiling and executing Test class?

A - Compilation error in Animal class

B - Vaccinating...

C - Compilation error in Test class

Java Class Design - 9

Consider the code of Test.java file:

```
class Player {
    String name;
    int age;

    Player() {
        this.name = "Sachin";
        this.age = 44;
    }

    public Object toString() {
        return name + ", " + age;
    }
}

public class Test {
    public static void main(String[] args) {
        System.out.println(new Player());
    }
}
```

What will be the result of compiling and executing Test class?

A - Compilation error

B - null, 0

C - Text containing @ symbol

D - Sachin, 44

Java Class Design - 10

Consider the code of Test.java file:

```
package com.training.ocp;

class Player {
    String name;
    int age;

    Player() {
        this.name = "Virat";
        this.age = 29;
    }

    public int hashcode() {
        return 100;
    }
}

public class Test {
    public static void main(String[] args) {
        System.out.println(new Player());
    }
}
```

Hexadecimal representation of 100 is 64.

Which of the following option is correct?

A - Code compiles successfully and on execution always prints “com.training.ocp.Player@64” on to the console

B - Code compiles successfully but throws an exception on executing it

C - None of the other options

D - Code doesn’t compile successfully

Java Class Design - 11

Consider the code of Test.java file:

```
class Player {
    String name;
    int age;

    void Player() {
        this.name = "Virat";
        this.age = 29;
    }

    public String toString() {
        return "Name: " + this.name + ", Age: " + this.age;
    }
}

public class Test {
    public static void main(String[] args) {
        System.out.println(new Player());
    }
}
```

What will be the result of compiling and executing Test class?

A - Name: Virat, Age: 29

B - An exception is thrown at runtime

C - Compilation error

D - Name: null, Age: 0

Java Class Design - 12

Consider the code of Test.java file:

```
class Calculator {
    public static void add(int x, int y) {
        System.out.println("The sum is: " + x + y);
    }
}

public class Test {
    public static void main(String[] args) {
        Calculator.add(15, 25);
    }
}
```

What will be the result of compiling and executing Test class?

A - The sum is: 1525

B - Compilation error

C - The sum is: 40

Java Class Design - 13

Consider below code:

```
//Test.java
abstract class Animal {
    abstract void eat();
}

class Dog extends Animal {
    public void eat() {
        System.out.println("Dog eats biscuit.");
    }
}

class Cat extends Animal {
    public void eat() {
        System.out.println("Cat eats fish.");
    }
}


public class Test {
    public static void main(String[] args) {
        Animal [] animals = new Dog[2];
        animals[0] = new Dog();
        animals[1] = new Cat();

        animals[0].eat();
        animals[1].eat();
    }
}
```

What will be the result of compiling and executing Test class?

A - Runtime exception

B - Dog eats biscuit.

C - Cat eats fish.

D - None of the other options.

E - Compilation error

Java Class Design - 14

Consider the code of Test.java file:

```
class Player {
    String name;
    int age;

    Player() {
        this.name = "Yuvraj";
        this.age = 36;
    }

    protected String toString() {
        return name + ", " + age;
    }
}

public class Test {
    public static void main(String[] args) {
        System.out.println(new Player());
    }
}
```

What will be the result of compiling and executing Test class?

A - Yuvraj, 36

B - null, 0

C - Text containing @ symbol

D - Compilation error

Java Class Design - 15

Consider below code:

```
class Animal {
    void eat() {
        System.out.println("Animal is eating.");
    }
}

class Dog extends Animal {
    public void eat() {
        System.out.println("Dog is eating biscuit.");
    }
}

public class Test {
    public static void main(String[] args) {
        Animal [] animals = new Dog[2];
        animals[0] = new Animal();
        animals[1] = new Dog();

        animals[0].eat();
        animals[1].eat();
    }
}
```

What will be the result of compiling and executing Test class?

A -

```
Animal is eating.
Animal is eating.
```

B -

```
Animal is eating.
Dog is eating biscuit.
```

C - Compilation error

D - Runtime exception

Java Class Design - 16

Consider the code of Test.java file:

```
abstract class Animal {
    public static void vaccinate() {
        System.out.println("Vaccinating...");
    }

    private abstract void treating();
}

public class Test {
    public static void main(String[] args) {
        Animal.vaccinate();
    }
}
```

What will be the result of compiling and executing Test class?

A - Compilation error in Test class

B - Compilation error in Animal class

C - Vaccinating...