

0. 단순 문제

```
public class Test1 {  
    public static void main(String[] args) {  
        //숫자를 55부터 150까지 더한 결과를 출력하세요.  
        //단, 11의 배수는 제외하고 더해주세요.  
    }  
}
```

//1. getter, setter 연습문제입니다.

```
class Problem1{  
    private int number;  
    private String name;  
  
    //이후 부족한 부분을 작성하세요.  
    public String getName() {  
        return name;  
    }  
    public int getNumber() {  
        return number;  
    }  
}
```



```
}  
  
public class P1 {  
    public static void main(String[] args) {  
        Problem1 p = new Problem1();  
  
        p.setName("Java"); //이 부분을 작성하세요.  
        p.setNumber(2); //이 부분을 작성하세요.  
  
        int sum =  //이 부분을 작성하세요.  
        System.out.println("(p객체에 저장되어있는 number) + 5의 결과: " + sum);  
        System.out.println("저장되어 있는 이름 = " + ); //이 부분을 작성하세요.  
    }  
}
```

1번. *getter, setter*

실행결과 :

(p객체에 저장되어있는 number) + 5의 결과: 7
저장되어 있는 이름 = Java

//2. 매개변수가 있는 생성자 예제 문제

class Problem2{ //매개변수가 있는 생성자

private int number1;

private int number2;

protected int Sum() {

return this.number1 + **this.number2**;

}

//이후 부분을 작성하세요.

}

public class P2 {

public static void main(String[] args) {

Problem2 p1 = **new** Problem2(2, 4);

Problem2 p2 = **new** Problem2(1, 2);

System.**out**.println();

System.**out**.println();

}

}

2번. 생성자

실행결과 :

6
3

//3.

//Java Program 출력하는 문제

//Python Program 출력하는 문제

```
public class P3 {  
    //부족한 부분을 완성하세요  
    private static void printLangName() {  
        System.out.println(s + " Program");  
    }  
  
    public static void main(String[] args) {  
        ("Java"); //이 부분을 완성하세요  
        ("Python"); //이 부분을 완성하세요  
    }  
}
```

3번. 인수와 매개변수

실행결과 :

Java Program
Python Program

//4. 클래스 변수 / 클래스 메소드 연습

```
class Problem4{
```

```
     = 2; //이 부분에서 변수 number를 정의 및 초기화하세요.
```

```
    static String notFour() {  
        if (number >= 4 ) {  
            return "4이상은 금지";  
        }  
        return "4미만은 가능";  
    }  
}
```

```
}
```

```
public class P4 {
```

```
    public static void main(String[] args) {  
        System.out.println(); //이 부분을 작성하세요  
        Problem4.number++;  
        Problem4.number++;  
        System.out.println(); //이 부분을 작성하세요  
    }  
}
```

4번. 클래스변수/클래스 메소드

실행결과 :

4미만은 가능
4이상은 금지

//5. 오버로딩 연습

```
class Problem5{  
    void   
        System.out.println("Test Version");  
    }  
    void {  
        System.out.println("Test Version:" + number);  
    }  
    void  {  
        System.out.println("Test Version:" + number + version);  
    }  
}  
public class P5 {
```

```
    public static void main(String[] args) {  
        Problem5 p = new Problem5();  
        p.TEST();  
        p.TEST(2);  
        p.TEST(2.51, "beta");  
    }  
}
```

5번. 오버로딩

실행결과 :

```
Test Version  
Test Version:2  
Test Version:2.51beta
```

6번. 익명 객체

```
class Annony{
    void override_method() {
        System.out.println("이 메소드는 오버라이딩 되어야 합니다");
    }
}
public class Test {
    public static void main(String[] args) {
        Annony new_annony;

        new_annony.override_method();
    }
}
```

실행결과 : 이 메소드는 오버라이딩 되었습니다!

7번. 생성자 오버로딩

실행결과 :

기본 생성자입니다.
p1은 기본 이름인 '아무개'와 '0'살을 갖습니다
p1의 이름: 아무개
p1의 나이: 0살
p1은 사용자가 지정한 이름인 '홍길동'과 '50'살을 갖습니다
p2의 이름: 홍길동
p2의 나이: 50살

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

    public Person() {
        this();
    }
}
```

```
public class Test {
    public static void main(String[] args) {
        Person p1 = new Person();

        System.out.println("p1은 기본 이름인 '아무개'와 '0'살을 갖습니다");
        System.out.println("p1의 이름:" + p1.name);
        System.out.println("p1의 나이: " + p1.age + "살");

        Person p2 = new Person("홍길동", 50);
        System.out.println("p1은 사용자가 지정한 이름인 '홍길동'과 '50'살을 갖습니다");
        System.out.println("p2의 이름:" + p2.name);
        System.out.println("p2의 나이: " + p2.age + "살");
    }
}
```


8. 각자 재량에 맡길게요

```
class AClass{
    void method_a() {
        System.out.println("This is method A");
    }
}

public class Test2 {
    public static void main(String[] args) {
        //여기서 변수 c를 사용하여 method_a()라는 메소드를 호출했을 때
        //"This is method B"가 나오도록 하세요.
        //단, AClass 내용은 수정하면 안됩니다.
        //어떤 방법으로 해도 상관없습니다. 대신 c변수를 사용해서 호출해야합니다
        AClass c;
    }
}
```

9. 각자 재량에 맡길게요

```
//child_method를 호출하세요.  
//어떤 방법을 사용해도 괜찮습니다.  
class ParentClass{  
    class InnerClass {  
        void child_method() {  
            System.out.println("parent");  
        }  
    }  
}  
  
public class Test3 {  
  
}
```

```

public class DBConnection {
    private Connection con;

    public DBConnection() {
        try {
            // 아래의 구문은 ?????를 하는 과정이다
            Class.forName("com.mysql.cj.jdbc.Driver");
            // DB와 커넥션을 시도한다
            this.con = DriverManager.getConnection( );
            System.out.println("연결완료");
        }
        catch (Exception e){
            System.out.println("오류 발생!!");
            System.exit(0);
        }
    }

    public Connection getConnection() {
        return con;
    }

    private void get_data() throws Exception {
        String sql = "SELECT id FROM WHERE userID = ?";
        PreparedStatement stmt = this.con.prepareStatement( );
        stmt.setString(1, );
        ResultSet set = stmt. ;
        System.out.println( );
    }

    public static void main(String[] args) {
        new DBConnection().getConnection();
    }
}

```

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

```

```

// JAVA를 사용하여 MYSQL의 Database에 접속해서 데이터를 가져오려고 한다
// 필요한건 tester 유저에 대한 ID의 조회이다. 아래의 내용을 채우시오.
// 현재 DB 서버의 IP 주소: 123.123.123.123
// 현재 DB 서버의 PORT: 3330
// 접속하려는 DB명: "TEST_DB", 유저명: "tester", 비밀번호: "1234"

```

11.

Stream Compare

```
import java.util.Arrays;
import java.util.List;

class PersonData{
    String name;
    int age;
    public PersonData(String name, int age) {
        this.name = name;
        this.age = age;
    }
}

public class ReplaceCode {
    public static void main(String[] args) {
        // for문을 사용하지 말고 아래의 지시를 따르시오. (Stream 등 사용)

        // 문제1. 1 ~ 9까지의 숫자를 더하세요

        // 문제2. 아래 array에서 짝수를 제외하고 출력하세요
        int[] array = new int[] {1,2,3,4,5,6,7,8,9,10};

        // 문제3. OBJECT가 저장되어있는 아래 list를 나이순으로 정렬하세요
        List<PersonData> pd = Arrays.asList(
            new PersonData("c", 13),
            new PersonData("javascript", 11),
            new PersonData("java", 2),
            new PersonData("python", 50));
    }
}
```