**문제 3번**

error: variable s might not have been initialized

System.out.print(s);

^

선언만 하고 정의는 하지 않을 경우, 출력해야 하는데 뭘 출력할지 모르겠음.

따라서 initialized 가 되지 않았다며 컴파일 오류 발생.

**문제 4번**

String은 기본 자료형이 아님. 참조 데이터형(p.61).

기본 자료형은 다음과 같다.

1) 정수형: ㄱ.byte ㄴ.short ㄷ.int ㄹ.long

2) 실수형: ㄱ. float ㄴ. double

3)문자형: char

4)논리형: boolean

**문제 5번**

문자형 자료형은 \u0000 ~ \uFFF (0~65535) 범위의 값을 입력할 수 있다(p.57).

문자형은 ‘a’, ‘ㄱ’, ‘Z’ 등의 문자를 입력할 수 있으며, 몇 가지 특수 문자도 가능함.

가능한 문자형에 입력 가능한 특수문자는 다음과 같다(p.61).

1)\\ 역슬래시 2)\n 줄바꿈 3)\r 캐리지 리턴

4)\b 백스페이스 5)\t 탭 6)\' 따옴표 7)\" 쌍따옴표

**문제 6번**

a. float foo = -1; 가능: 우변은 int형인데, float형이 int형보다 더 크므로 가능.

b. float foo = 1.0; 에러: 우변이 자동으로 double형으로 인식. float은 double보다

작으므로 에러

c. float foo = 42e1; 에러: 유효숫자와 거듭제곱꼴의 곱으로 나타내는 것은 double형.

d. float foo = 2.02f; 가능: 그냥 두면 double형으로 인식하지만 끝에 f가 붙어서 float형

e. float foo = 3.03d; 에러: 그냥 둬도 double인데 끝에 d까지 붙여서 빼박 double형.

f. float foo = 0x0123; 가능: 0x는 단순히 수를 나타내는 진법만 바꾸는 것 뿐이고,

int형과 마찬가지이므로, a처럼 가능.

**문제 7번**

ㄱ. boolean b = 0; 뒤에 int형이라 에러.

ㄴ. boolean b = true; 논리형은 true or false 값을 입력. 가능.

ㄷ. boolean b = "true"; 큰따옴표로 인해 string으로 인식. 에러.

ㄹ. boolean b = 'true'; 작은따옴표로 인해 char로 인식. 에러.

ㅁ. boolean b = (3>=0); 괄호 안의 논리식에 따라 true or false를 입력. 가능.

**문제 9번**

ㄱ. float a = 1; //float은 int보다 큰 자료형이므로 가능.

ㄴ. float b = 1L; //float은 long보다 큰 자료형이므로 가능.

ㄷ. float c = 1.0F; //1.0은 doule형이지만 뒤에 F 붙여서 float으로 인식.

ㄹ. float d = 1.0; //1.0이 double형으로 인식, float이 더 작으므로 에러.

**문제 10번**

System.out.println("" + 2 + 3); => 23

// ‘+’ 기준으로 한 쪽이라도 문자열일 경우, 다른 쪽도 문자열‘화’ 되어버림 ->

2가 문자열로 인식됨 -> 앞에 문자열이 하나라도 등장할 경우 연쇄 작용으로

그 뒤는 모두 문자열로 인식됨. (단, 괄호를 이용해 일부는 계산 가능)

System.out.println(2 + 3); => 5

System.out.println(2 + 3 + ""); => 5

//앞에부터 연산 진행하므로, 왼쪽의 ‘+’ 연산자 기준으로 양쪽은 int로 인식,

뒤쪽의 ‘+’ 연산자에는 뒤쪽에 문자열이 있으므로 둘 다 문자열로 인식.

System.out.println(2 + "" + 3); => 23

//이건 빼박 문자열 인식 -> int 그대로 출력

**문제 12**

boolean a = true & true; 가능.

int a = 4 & 5; int로 인식

int c = 0 > 3; boolean이라 안 됨.

float d = 40.0; double을 float에 넣으려니 안 됨. double이 더 큼

boolean a = (boolean) 1; int로 인식. 안 됨.

**문제 13번**

boolean res = ++a==1 || ++b==1;

(a + " " + b + " " + res)

res= 이후 ++a==1을 만족하므로 res에 속한 다음 연산은 수행하지 않고,

바로 그 이후 연산으로 넘어감. 연산의 최소화를 위해.

따라서 a는 ++a를 연산했으므로 1이고,

그 뒤에 있는 ++b는 지나가지 못하므로 그대로 0임.

“1 0 true”가 출력됨.