1.잘못된 문장을 골라내고 그 이유를 설명하시오.  
public class DataEx01 {  
  public static void main(String[] args) {  
    char   a='a';       //A.  
    char   b="a";       //B.  
    String  c="a";       //C.  
    String  d='a';       //D.  
    **char   e='ab';      //E.**    String  f="ab";      //F.    
 }  
}

**이유: char 형태의 변수는 어떤 문자든 한글자만 입력해야합니다.**

**char형의 변수의 범위는**

**부호가 있을때는 -128 ~ 127,**

**부호가 없을때는 0 ~ 255 까지를 저장할 수 있습니다.**

**이 char형 변수에 'a' 혹은 '1' 라고 입력을 하면**

**문자 a의 아스키 코드 값 또는 숫자 '1'의 아스키 코드 값을 나타내게됩니다.**

**(예를 들어서 a의 아스키 코드값은 97입니다)**

**그리고 '  ' 안에는 한 글자만 들어가야합니다.**

**"  "안에는 여러 글자가 들어가지만 '  '는 char형 변수에서 쓰이기 위해 한글자만 들어가게 설계되었습니다.**

2.잘못된 문장을 골라내고 그 이유를 설명하시오.  
public class  DataEx02{  
  public static void main(String[] args)   {  
**byte var1=128; //A.**  
short var2=128; //B.  
**int var3=28L;//C.**  
long var4=128L; //D.  
**float var5=123456.789123;             //E.**  
     double var6=123456.789123;            //F.    
  }  
}

**이유**

**A: byte는 -128 ~ 127 까지의 숫자만 입력시킬 수 있습니다. 128은 127보다 큰 수 이기때문에 에러입니다.**

**C: 숫자 뒤에 "L"을 붙이면 그건 long형 숫자입니다. 그렇기때문에 int에 long형 숫자를 넣어서 에러입니다.**

**E: float형 변수에는 float형 숫자를 입력해줘야합니다. float형 숫자는 숫자 뒤에 "f"를 붙여줘야합니다. (예: 10.0f)**

3.잘못된 문장을 고쳐서 에러가 발생하지 않도록 수정하시오.  
001:public class DataEx03{  
002:  public static void main(String[] args) {  
003:    byte var1=127;  
004:    short var2=128;  
005:    int var3=128;  
006:    long var4=128L;  
007:    var4 = var1;  
008:    System.out.println(var1+ ", " + var2);  
**009:    var1 = var3;**  
010:    System.out.println(var1+ ", " + var3);  
011:  
**012:    float var5=123456.789123;**  
013:    double var6=123456.789123;  
**014:    var5 =var6;**  
015:    System.out.println(var5+ ", " + var6);  
016:    var6 = var5;  
017:    System.out.println(var5+ ", " + var6);  
018:  }  
019:}

**이유**

**009: var3는 int형이고 var1은 byte형인데요, byte형에 int형을 넣으려면, byte의 그릇이 더 작기때문에 캐스트(cast)를 해주셔야합니다. (해결방법: var = (byte)var3; )**

**012: var5는 float형인데, 입력되려는 숫자는 double형 숫자입니다. 숫자 뒤에 f를 붙여서 float형 숫자로 입력시켜줘야합니다. (혹은, 숫자 앞에 (float)이라고 캐스트 해줘도 됩니다. (float)(123456.789123) )**

**014: var5는 float형이고 var6는 double형입니다. 이 경우에 캐스트를 해줘야합니다 (해결방법 var5 = (float)var6); )**

4.다음 프로그램을 컴파일 했을 때 결과를 추측해보시오.  
public class DataEx06 {  
  public static void main (String[] args) {  
    byte b = 127;  
    byte c = 126;  
    byte d = b + c;  
  }  
}

**결과 : 오류가 납니다**.

**이유:**

**byte변수의 범위는 -128부터 127까지입니다.**

**b는 -128 ~ 127 범위 안에 들어가기때문에 합당합니다**

**c도 -128 ~ 127 범위 안에 들어가기땜누에 합당합니다.**

**하지만 b+c는 253인데, 이 숫자는 byte형 변수의 범위에 들어가지 않습니다.**

**그렇기때문에 오류가 납니다.**

5.다음 실행 결과를 예측하시오.  
class ChoiceEx05 {  
  public static void main (String []args) {  
    Float f = new Float("12");  
**switch (f) {**  
      case 12: System.out.println("Twelve");  
      case 0: System.out.println("Zero");  
      default: System.out.println("Default");     
    }  
**System.out.println("j="+j);**  
  }  
}

**결과: 오류가 납니다.**

**이유:**

**첫째: switch 함수 안에는 int형 숫자만 들어가야합니다. (정수만 들어가야함)**

**그런데 float형 숫자를 넣어서 오류가 난겁니다.**

**둘째: 이 코드 상에 j는 선언/초기화 되지 않았기때문에 오류가 납니다.**

6.다음 실행 결과를 예측하시오.  
class ChoiceEx06 {  
  public static void main (String []args) {  
    int x=20;  
    int odd = x%2;  
**if (odd) {**      System.out.println("odd");  
    } else {  
      System.out.println("even");  
}  
  }  
}

**결과: 에러가 납니다.**

**이유:**

**if 문에서 조건을 적을때,**

**숫자를 쓰면, 꼭 부등식을 써줘야합니다.**

**예를 들어서**

**if 문에 조건으로 10을 꼭 써야한다고 하면**

**if( 10 > 0 )**

**이런식으로 부등식을 써줘야합니다.**

해결방법:

class ChoiceEx06 {  
  public static void main (String []args) {  
    int x=20;  
    int odd = x%2;  
**if (odd == 1) {**      System.out.println("odd");  
    } else {  
      System.out.println("even");  
    }  
  }  
}

**(int odd = x%2 구문에서 x의 값을 2로 나눈 나머지 값을 입력시키기때문에, 그 값이 1이면 홀수, 0이면 짝수입니다)**

7.다음 결과를 예측하시오.  
class ChoiceEx09 {  
  public static void main (String []args) {  
    int i = 1;  
    int j = i++;  
    if ((i>++j) && (i++ ==j)) {  
      i +=j;  
}  
    System.out.println("i = " + i + ", j = " + j);  
  }  
}

단계단계의 결과 분석:

int i = 1;

**i = 1**

    int j = i++;

**j=1,  i는 j에게 1 값을 준 이후에 자기 자신에게 1을 더해준다. 그렇기때문에 i = 2**

    if ((i>++j) && (i++ ==j)) {

**++j에 의해서 우선적으로 j에 1을 더한 후에 i>j 조건을 실행한다. (i=2, j=2,  2>2 는 false)**

**2>2라는 조건이 성사되지 않기때문에 && 이후의 조건은 실행되지 않는다.**

**결국 i = 2, j= 2**

**한번 훑어보시고**

**잘 이해안되시거나 더 궁금한 점 있으시면 쪽지 주세요**

**성심성의껏 답해드리겠습니다.**

**그러면, 즐거운 프로그래밍 공부 되세요~**