**Computer Language 2023**

**Assignment #1**

**Due: 12/Mar 23:59:59**

**Student ID:\_\_21102054\_\_ Name: \_\_Lim Sae-Yeon\_\_**

**1. Provide at least 5 examples of invalid variable names. Also, explain why such variable names cannot be used. You cannot use the examples presented in the lecture slide.**

**1. double sae-yeon;**

**: ‘-‘ is special characters except ‘$’ and ‘\_’**

**2. int 23ITM;**

**: start with digits ‘2’**

**3. double while;**

**: “while” is Java Keywords**

**4. int hello world;**

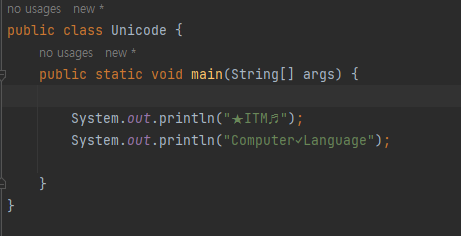
**: A whitespace exists between “hello” and “world”**

**5. double null;**

**: Null literals cannot be used as variable names**

**2. Write a program to print the following string using UNICODE literals (only for symbols). Capture your source code, the output of your program and paste them in this document.**

**[ Source Code ] [ Answer ]**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**3. Execute the following program. What is the output of this code? Explain why/how such a result was calculated.**

**public class Hello {  
 public static void main(String[] args) {  
  
 int intValue = 12345;  
 System.*out*.println((byte) intValue);  
   
 }  
}**

**[ Answer ]**

**57 (decimal) = 11000000111001 (bin)**

**00000000 00000000 00110000 00111001 (4 byte)**

**00111001 (after casting, bin) = 57 (decimal)**

**Converting 12345 to binary is 11000000111001.**

**The int type is a size of 4byte, so it is expressed as 00000000 00000000 00110000 00111001.**

**However, in the process of casting as byte type,**

**only 00111001 is left as the byte size assigned becomes 1 byte. This is 57 in decimal.**

**4. How to declare a variable for the following case? Write your answer here.**

**- int type variable ‘height’**

**[Answer] int height;**

**- double type variable ‘size’ initialized to 0.25**

**[Answer] double height = 0.25;**

**- double type variable ‘total’ initialized with the sum of the values of ‘height’ variable and ‘size’ variable**

**[Answer] double total = height + size;**

**- char type variable ‘c’ initialized with ‘a’**

**[Answer] char c = ‘a’;**

**- double type constant ‘BODYTEMP’ initialized to 36.5**

**[Answer] final double BODYTEMP = 36.5;**

**5. what is the result and type of the following statements? Explain the reason as well.**

**a) 67+12.8**

**>> 79.8 (Double)**

**Reason : 67 is int type because all integer value is assigned to int type basically. 12.8 is floating point type and double type. Java convert small type value to larger type when it hav to do arthmetic operation with different types of values. So, in this case, int type which is smaller is transformed to double which is larger. As a result, the result value is 79.8, double type.**

**텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명**

**b) ‘c’+1**

**>> 100 (Int)**

**Reason : ‘c’ is same as 99 in decimal ascii table. The unicode number of it is 0063 in hexademical. Char type has 2 bytes and int has 4bytes, so char type should be converted to int type. 0063 in hexademical is as same as 99 in decimal. As a result, the type of result value is 100, which is int type.**

**텍스트이(가) 표시된 사진

자동 생성된 설명**텍스트이(가) 표시된 사진

자동 생성된 설명

**c) 10/3**

**>> 3 (Int)**

**Result : Two values are int types, so the result value must be int type. So the quotient is only output in integer calculation, without floating point.**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**d) 10.0/3**

**>>** **3.3333333333333335 (Double)**

**Reason : 10.0 is double type whereas 3 is int type. The types are different, so the type of result value is bigger one, which is double type because in arithmetic operations, smaller types are converted to larger types.**

**텍스트이(가) 표시된 사진

자동 생성된 설명 텍스트이(가) 표시된 사진

자동 생성된 설명**

**e) 10 == 9**

**>> False (Boolean)**

**Reason : There is “==”. This mean the computer should show whether the statements is true or false. As a result, a result type should be Boolean type to show true or false. And 10 is not eqult to 9, result is “False”.**

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

**7. Execute the following codes. Then, you will see eight values printed on the console. For each value, explain how it was calculated.**

public class Ex07 {  
 public static void main(String[] args) {  
 byte b = 127;  
 int i = 100;  
  
 System.*out*.println(b + i);  
 System.*out*.println(10 / 4);  
 System.*out*.println(10.0 / 4);  
 System.*out*.println((char) 0x12340041);  
 System.*out*.println((byte) (b + i));  
 System.*out*.println((int) 2.9 + 1.8);  
 System.*out*.println((int) (2.9 + 1.8));  
 System.*out*.println((int) 2.9 + (int) 1.8);  
  
 }  
}

**[ Answer ]**

1) System.*out*.println(b + i);

>> 227 (Int)

Reason : The type of b, byte is converted to int type. Value of b is 127 despite of the converting. It because byte is smaller than int type.

**텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명**

2) System.*out*.println(10 / 4);

>> 2 (Int)

Reason : Operands are int type, so the type of output should be int.

**텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명**

3) System.*out*.println(10.0 / 4);

>> 2.5 (Double)

Reason : There are different types of value, double which is bigger and type which is smaller. So, the type of 4 will be converted to double type. And the result will be double type including floating-point.

**텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명**

4) System.*out*.println((char) 0x12340041);

>> A (Char)

Reason : 0x12340041 is 0001 0010 0011 0100 0000 0000 0100 0001 in binary. The char type has 2 bytes, so 0000 0000 0100 0001 is assigned to (char) 0x12340041. It is 41 in heademical, which mean ‘A’ in ascii table.

**텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명**

5) System.*out*.println((byte) (b + i));

>> -29 (Byte)

Reason : 127 is maximum value in byte type. The binary begins with 1 is smaller than zero. When 1 is added, the intermediate result is 128, which is the minium value of byte type. Add 99 to the result, then the final result bcomes -29.

**텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명**

6) System.*out*.println((int) 2.9 + 1.8);

>> 3.8 (Double)

Reason : 2.9 became 2.9 while casting to int type. And then, it converted to double type to do a calculation with 1.8. So the result should be double type.

**텍스트이(가) 표시된 사진

자동 생성된 설명**

7) System.*out*.println((int) (2.9 + 1.8));

>> 4 (Int)

Reason : 2.9 + 1.8 is 4.7, which is double type. However, there was a casting to int type from double type, so the result should be int type without expressing floating point.

**텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명**

8) System.*out*.println((int) 2.9 + (int) 1.8);

>> 3 (Int)

Reason : Both values are double type, however, there are castings each value, the values became int type. And then, the computer will perform calculatons with two int type values. So the result should be int type.

**텍스트이(가) 표시된 사진

자동 생성된 설명** **텍스트이(가) 표시된 사진

자동 생성된 설명**