Data types

Which of the following data types can only store data approximately?

Values of all types but float and double are stored accurately

Select one:

- a. String
- b. long
- c. boolean
- d. int
- e. float ✔

Will the following piece of code compile? If not which line causes the compile error?

The line numbers are just for reference.

```
1   int no=10;
2   boolean flag=true;
3   double rate=1.5;
4   String unit="1051";
5   System.out.println(no+unit+rate);
6   System.out.println(no+flag+unit+rate);
```

output: 1010511.5

Select one:

- a. no, 1
- b. no, 5
- c. yes
- d. no, 2
- e. no, 4
- f. no, 6 ✔
- g. no, 3

If int concatenated with boolean without string between them, Java will calculate int+boolean

Data type conversions

Will the following statement sequence compile? If not which line causes the compile error?

The line numbers are just for reference.

```
1  double i;
2  int j = 2;
3  i = j;
4  j = i;
```

i=2.0

Select one:

- a. No, line 4 ✔
- b. No, line 3
- c. No, line 1
- d. No, line 2
- e. Yes

convert double (8 bytes) to int (4 bytes): bigger data type to smaller data type (narrowing conversion)

explicit cast (int) required: target data type enclosed in parenthesis

What is wrong with the following statement?

```
double no1=10,n2o=3.6,no3;
```

int 10->double no1:
automatic conversion

Select one:

- a. Nothing is wrong. The statement is syntactically correct ✔
- b. We cannot separate the variables by a comma ','
- c. no3 must initialized for the statement to get compiled.
- d. no1 must be initialized with a double value
- e. n2o variable name is incorrect

Your answer is correct.

Variable & constant

Can a Java constant that has been declared and initialised be used to initialise a Java variable?

Can a Java variable that has been declared but not yet initialised be used to initialise a Java constant?

Select one:

a. No, No

b. Yes, No ✔

c. No, Yes

d. Yes, Yes

```
final double CONSTANT =3.5;
      int variable= (int)CONSTANT;
      System.out.println(variable);
```

output: 3

```
int i=10;
final int CONSTANT=i;
System.out.println (CONSTANT);
```

output: 10

Your answer is correct.

Which of the following is NOT true for both Java variables AND Java constants?

Hint: If you are unsure try it in IntelliJ. This applies to all questions in this pre-reading quiz.

Select one:

a. Neither can be used without first being declared

b. Both can be initialised at declaration

c. Both must have a data type

d. Both can be initialised after declaration

e. Both can have a name that begins with a number ✔
Java identifiers: first character can not be a digit

f. If you think all of the above are true for both Java variables and Java constants, select this option

Rules for naming variables:
1. must begin with a letter of alphabet,(or_,or $)
2. cannot use a java keyword (eg.public, class, static, void)
but public1 is ok
3. 1st letter: lowercase, 1st letter of subsequent words: UPPERCASE

Rules for naming constants:
use all UPPERCASE letters with _ separating the words (eg. FIRST_NAME)

Variable

Given the following syntax template:

dataType *variableName* [, *variableName* [= *dataValue*]]...;

Would it be valid to declare a variable with an initial value and a variable without an initial value in the same declaration statement?

```
int a, b=10;
a=5;
System.out.println(a+b);

output: 15
```

Select one:

- a. No
- b. Yes without restrictions
- c. Yes with restrictions ✔

These variables must be the same data type

Your answer is correct.

Will the following statement sequence compile? If not which line causes the compile error?

The line numbers are just for reference.

```
1 int i;
2 int j = i;
3 j = 3;
```

variable i must be initialised before it is used

Select one:

- a. yes
- b. no, line 1
- c. no, line 3
- d. no, line 2 ✔

Assignment statement

What is the output of the following piece of code?

```
int x,y,z;
x=3;
y=5;
z=7;
z=x=y;
System.out.println(x+" "+ y +" "+ z);
```

Select one:

- a. 3 5 7
- b. 5 7 5
- c. 5 5 5 ✔

the value of y is assigned to x: x=5
the value of x is then assigned to z: z=5

- d. 15 15 15
- e. 3 3 3
- f. 5 3 7
- g. 7 7 7

Comment

Which of the following starts a single line Java comment?

Select one:

- a. // ✔

multi-line comment:
/*
*/

- b. //*
- c. */
- d. ///
- e. If you think none of the above starts a single-line Java comment, select this option