

**GROUP - A** <10 points >

---

**Marking Guideline**

For each question, give +0.5 if correct, -0.5 if wrong, and 0 if blank. If the overall total is negative, record it (on the test's cover sheet) as zero.

- T** Parameters in Java are passed by value
  - T** A method can return at most one thing
  - T** A package can contain more than one class
  - T** All classes in a package can be imported by using only one import statement
  - F** If a variable is used before being declared, *MM* will generate an error
  - F** The program that executes *bytecode* is called `javac`
  - T** `public` is a reserved word in Java
  - T** Two methods in a class can have the same name if they have different signatures
  - F** Two methods in a class can have the same signature if they have different return types
  - F** Two methods in a class are said to be overloaded if they have the same signature
  - F** Two fields in a class can have the same name if they have different types
  - F** The expression  $a+b*c$  is the same as  $(a+b)*c$
  - T** The expression  $a/b*c$  is the same as  $(a/b)*c$
  - T** The expression  $a-a+b-b == 0$  is always `true`
  - F** An escape sequence is used to override automatic promotion.
  - F** Java is case-insensitive
  - F** If `x` is an `int` and the expression  $x/2$  evaluated to an integer, then `x` must be even
  - T** If `x` is an `int` and the expression  $x\%2$  evaluated to 1, then `x` must be odd
  - T** If the postcondition of a method is not met, blame its implementer
  - F** The compiler detects all logic errors in an app
-

**Marking Guideline**

Give 0 or 1 for each output. In case of error, give +1 for indicating the error type and +1 for a correct description of the error.

B.1    `int a = 3;  
int b = 28;  
a = a - 6 % 3;  
b = b / 5;  
IO.println(a);  
IO.println(b);`

3

5

B.2    `int x = 20;  
final int YR = 5;  
x = x % 3;  
YR = YR / 2;  
IO.println(x);  
IO.println(YR);`

Syntax error in 4th line: attempting to modify a final.

B.3    `int a = 30;  
boolean m = "true";  
IO.println(a + 1);  
IO.println(m);`

Syntax error in the 2nd line: LHS is a boolean, RHS is a String

## GROUP - B, *continued*

---

```
B.4 byte a = 120;
    boolean g = a != 0;
    byte b = (byte) (a + 9);
    IO.println(g);
    IO.println(b);
```

true

-127

```
B.5 short a = 530;
    short b = a;
    IO.println(a + 1);
    IO.println(b - 1);
```

531

529

```
B.6 byte x = 12;
    char y = 'x';
    IO.println(x - 1);
    IO.println(y - 1);
```

11

119 *(It is not 'w' because the -1 will auto-promote to int. To get 'w', cast to char)*

```
B.7 int x = -5;
    double y = -5 / 2 * 2;
    IO.println(x % 2);
    IO.println(y);
```

-1

-4.

**GROUP - B, continued**

---

```
B.8  double base = 5.5;
      double height = 4.0;
      double area = (1/2) * base * height;
      double hypo = Math.pow(height, 2);
      IO.println(area);
      IO.println(hypo);
```

0.  
16.0

```
B.9  float mass = 0.5;
      float weight = 9.8 * mass;
      IO.println(mass / 2);
      IO.println(weight);
```

Syntax error in the 2nd line: attempting to assign a double to a float.

```
B.10 double x = 11.2;
      IO.println(((int) x) / 2);
      IO.println(Math rint(x) / 2);
```

5  
5.5

Consider the following API:

type.jba

## Class Sint

This class offers services for manipulating sints. A sint is a special kind of integer; it is a fictitious term that has been made up solely for this test.

Field Summary	
static int	<a href="#">MAXI</a> The highest allowed value for a sint

Method Summary	
static double	<a href="#">average</a> (int a, int b) Returns the weighted average of the passed sints.
static double	<a href="#">average</a> (long a, long b) Returns the biased average of the passed sints.

The following app, UseSint, uses the Sint class:

```
1  import type.lang.*;
2  import Sint.*;
3
4  public class UseSint
5  {  public static void main(String[] args)
6      {  //-----Prompt, Read, and Validate
7          IO.println("Enter the first sint value:");
8          int h = IO.readInt();
9          SE.require(h < MAXI);
10         //-----Prompt, Read, and Validate
11         IO.println("Enter the second sint value:");
12         int k = IO.readInt();
13         boolean ok = k <= MAXI;
14         //-----Compute averages as per API
15         double w = average(h, k);
16         double b = (long) (average(h, k));
17         //-----Output averages
18         IO.print("Their weighted average is: ");
19         IO.println(w);
20         IO.print("And their biased average is: ");
21         IO.println(b);
22     }
23 }
```

**C.1** <10 points >

---

**Marking Guideline**

Give +1 for indicating the error type and +1 for a correct description of the error.

Any five of the following:

<b>Line</b>	<b>Type</b>	<b>Description</b>
2	Syntax	Must specify a package: <code>type.jba.*</code> or <code>type.jba.Sint</code>
9	Syntax	MAXI is referenced as if it is a local variable!
9	Logical	The relational operator <code>&lt;</code> should be <code>&lt;=</code>
13	Syntax	MAXI is referenced as if it is a local variable!
13	Logical	The second input, <code>k</code> , was not validated!
15	Syntax	Cannot invoke <code>average</code> without the dot operator!
16	Syntax	Cannot invoke <code>average</code> without the dot operator!
16	Syntax	Casting the return does not change the invoked method: the weighted average is computed, not the biased one.

## C.2 <20 points >

---

Rewrite the UseSint app. It is supposed to prompt for and read two sint values from the user, validates them, and then output their weighted and biased averages.

```
1  import type.lang.*;
2  import type.jba.*;
3
4  public class UseSint
5  {  public static void main(String[] args)
6      {  //-----Prompt, Read, and Validate
7          IO.println("Enter the first sint value:");
8          int h = IO.readInt();
9          SE.require(h <= Sint.MAXI);
10         //-----Prompt, Read, and Validate
11         IO.println("Enter the second sint value:");
12         int k = IO.readInt();
13         boolean ok = k <= Sint.MAXI;
14         SE.require(ok);
15         //-----Compute averages as per API
16         double w = Sint.average(h, k);
17         double b = Sint.average((long) h, k);
18         //-----Output averages
19         IO.print("Their weighted average is: ");
20         IO.println(w);
21         IO.print("And their biased average is: ");
22         IO.println(b);
23     }
24 }
```

Note that in line 16 you can cast either or both parameters to long in order to force a binding with the biased average method.

### Marking Guideline

Give 20, 15, 10, 5, or 0. A few minor errors lead to 15, one major error leads to 10, and two or three major errors lead to 5.

## GROUP - D <40 points>

---

### D.1 <10 points >

Write an app that prompts for and reads an `int` from the user. If the input is not positive, the app terminates with the error message "Invalid Input". Otherwise, the app must round up the input to the nearest multiple of 25 and output the result. For example, if the input were:

10, 20, 25, 30, 60, 90, or 100

then the corresponding output would be (respectively):

25, 25, 25, 50, 75, 100, or 100

Note that if the input were already a multiple of 25, the output would be equal to it. Note also that even if your algorithm is incorrect, you will still receive partial marks for the rest of the app.

```
import type.lang.*;

public class App
{   public static void main(String[] args)
    {
        IO.print("Enter the integer ... ");
        int entry = IO.readInt();
        SE.require(entry > 0, "Invalid Input");

        int rounded = (entry + 24) / 25 * 25;

        IO.println(rounded);
    }
}
```

### **Marking Guideline**

Give 10, 7, 5, 3, or 0. A few minor errors lead to 7, one major error leads to 5, and two or three major errors lead to 3.

## D.2 <10 points >

---

Write an app that prompts for and reads a `double` from the user. If the input is not in  $(0,1000]$  (i.e. greater than 0 and less than or equal to 1000), the app terminates with the error message "Invalid Input". Otherwise, it implements the following algorithm:

- Look only at the first digit to the right of the decimal point
- If that digit is 2 or less, drop all decimal digits
- If that digit is 3 or higher, drop all decimal digits and increment the integer part by 1
- Output the integer part as an `int`

For example, if the input were 15.245, the above algorithm outputs 15, whereas if the input were 15.345, the output would be 16. Note that we drop the decimal part in both cases but the integer part is incremented only if the first decimal is 3 or more.

Note that even if your algorithm is incorrect, you will still receive partial marks for the rest of the app.

```
import type.lang.*;

public class App
{   public static void main(String[] args)
    {
        IO.print("Enter the double ... ");
        double entry = IO.readDouble();
        SE.require(entry > 0, "Invalid Input");
        SE.require(entry <= 1000, "Invalid Input");

        int rounded = (int) (entry + 0.7);

        IO.println(rounded);
    }
}
```

### **Marking Guideline**

Give 10, 7, 5, 3, or 0. A few minor errors lead to 7, one major error leads to 5, and two or three major errors lead to 3.

### D.3 <20 points >

---

Write an app that reads the  $x$ - $y$  coordinates of the centre of a circle and its radius. It then outputs *true* if the circle passes through the origin, and outputs *false* if it doesn't. In other words, the output is a boolean. Here is the algorithm of the sought app:

- Reads  $x$ ,  $y$ , and  $r$  from the user (all three are double's).  
The centre is at  $(x, y)$  and the radius is  $r$ .
- Compute the distance  $d$  between the centre and the origin.  
This distance is given by the square root of:  $x^2 + y^2$ .
- In math, where precision is infinite, the circle passes through the origin if  $d$  and  $r$  are equal. In computing, round off errors are inevitable so we modify this step as follows:  
If the absolute value of the difference between  $d$  and  $r$  is less than some very small number EPSILON, we say that the circle passes through the origin; otherwise, we say that it doesn't. For this problem, we take EPSILON to be  $1.E-10$ .

```
import type.lang.*;

public class App
{
    public static void main(String[] args)
    {
        final double EPSILON = 1.e-10;

        IO.print("Enter the centre's x ... ");
        double x = IO.readDouble();
        IO.print("Enter the centre's y ... ");
        double y = IO.readDouble();
        IO.print("Enter the radius ... ");
        double r = IO.readDouble();

        double d = Math.pow(x*x + y*y, 0.5);
        boolean passes = Math.abs(d - r) < EPSILON;

        IO.println(passes);
    }
}
```

#### Marking Guideline

Give 20, 15, 10, 5, or 0. A few minor errors lead to 15, one major error leads to 10, and two or three major errors lead to 5.