

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE AND ENGINEERING
APS105 — Computer Fundamentals
Quiz #1 Solutions
February 5, 1999, 4-5 p.m.

Questions #1 and #2 refer to the Java program below:

```
class Hello
{
    public static void main (String[] args)
    {
        System.out.println ("Hello world!");
    }
}
```

Question 1 [5 marks]

If this program is compiled and executed, what will it print?

Hello world!

Question 2 [5 marks]

To be compiled and executed the program must be stored in a file. What should the name of the file be? What is the UNIX command sequence for compiling and executing the program?

For this example:

Filename: Hello.java
javac Hello.java
java Hello

In general:

Filename: *ClassName*.java
javac *ClassName*.java
java *ClassName*

1

Question 3 [5 marks]

Assume that in your home directory you have a directory called `assign1` and that `assign1` contains a file called `Test.java`. Give a sequence of UNIX commands to do the following:

1. set your working directory to your home directory;
2. create a directory called `assign2` in your home directory;
3. copy the file from `Test.java` in `assign1` to `assign2`;
4. list the contents of `assign2`.

```
cd ~
mkdir assign2
cp assign1/Test.java assign2
ls assign2
```

Question 4 [5 marks]

What are the 3 essential components of a computer system? Briefly describe the role of each component (about 5 words each).

Central Processing Unit	brains, executes instructions, computes arithmetic (e.g. Intel Pentium processor)
Input/Output	ways for computer to interact with its environment (e.g. monitor, keyboard)
Memory System	stores programs (instructions), data (e.g. RAM, hard drive)

For questions #5 to #10, indicate what the Java program will print if it is compiled and executed.

Question 5 [5 marks]

```
class A
{
    public static void main (String[] args)
    {
        System.out.print ("I");
        System.out.println ("t's true that");
        System.out.println (" 1");
        System.out.print (" +");
        System.out.println ( 1);
        System.out.println (" --");
        System.out.print (" 2");
        System.out.println ("Right?");
    }
}

It's true that
1
+1
--
2Right?
```

2

Question 6 [5 marks]

```
class B
{
    public static void main (String[] args)
    {
        System.out.println ((6-15/3)+9*2/3/2);
        System.out.println (8/3);
        System.out.println (3.5/2);
    }
}
```

- 4
- 2
- 1.75

Question 7 [5 marks]

```
class C
{
    public static void main (String[] args)
    {
        double x;
        x = 8.5;
        System.out.println (Math.floor(x)/2);
    }
}
```

- 4.0

3

Question 8 [5 marks]

```
class D
{
    public static void main (String[] args)
    {
        int a,b;
        a = 23;
        b = 7;
        System.out.print ("The answer is ");
        if (a % b > 3);
        if (a != 3 && b == 7)
            System.out.println ("yes");
        else
            System.out.println ("no");
    }
}
```

The answer is yes

Question 9 [5 marks]

```
class E
{
    public static void main (String[] args)
    {
        int i;
        for (i=0; i < 5; i++);
        System.out.println("I must not forget semicolons.");
        System.out.println("I did not forget.");
    }
}
```

I must not forget semicolons.
I did not forget.

4

Question 10 [5 marks]

```
class F
{
    public static void main (String[] args)
    {
        int i = 0;

        while (i < 10) {
            System.out.print (i);
            if (i == 4 || i == 9)
                System.out.println ();
            i++;
        }
    }
}

01234
56789
```

Question 11 [5 marks]

Rewrite the program in question #10 using a do loop instead of a while loop

```
class G
{
    public static void main (String[] args)
    {
        int i = 0;

        do {
            System.out.print(i);

            if (i == 4 || i == 9)
                System.out.println();

            i++;
        } while (i < 10);
    }
}
```

Question 12 [5 marks]

In the following Java program, there are between 5 and 10 errors. Find as many as you can and correct them in the *simplest* way you can think of.

```
1. class H {
2.     public static vliod main (String[] args) {
3.     }
4.     int y;
5.     int a = StdIn.getint(); b = StdIn.getint();
6.     double c = Math.max(StdIn.getDouble(),b);
7.     for( y=0; y<c; y+=Math.sqrt(a) )
8.     {
9.         if( a < y && y < c )
10.            if( b < a i= a == 3 )
11.                System.out.sqr(c);
12.            else c -= a;
13.            else c += b;
14.            Math.Floor( y );
15.        }
16.    }
17. }
```

Answer

line 2 the word 'void' is misspelled 'vliod'
line 2,3 two opening braces '{' for main(). This is not an error by itself, except that 'class H' now misses its '}'
line 4 on line 7, y is used as a double (adding Math.sqrt()), so change y to a double
line 6 parentheses forgotten in the call to 'StdIn.getDouble()'
line 9 compound condition expressed incorrectly
line 10 type error because != and == have same precedence
line 11 no such method 'System.out.sqr(double)'
line 14 The result of Math.Floor(y) is not used; assign the result to a variable, or remove it

```
1. class H {
2.     public static void main (String[] args)
3.     {
4.         double y;
5.         int a = StdIn.getint(), b = StdIn.getint();
6.         double c = Math.max(StdIn.getDouble(),b);
7.         for( y=0; y<c; y+=Math.sqrt(a) )
8.         {
9.             if( a < y && y < c )
10.                if( b < a i= (a == 3) )
11.                    System.out.println( Math.sqrt(c) );
12.                else c -= a;
13.                else c += b;
14.                // Math.Floor( y );
15.            }
16.        }
17. }
```

Question 13 [5 marks]

Consider the following Java variable declarations:

```
int i;           // 32 bits of storage
double x;       // 64 bits of storage
boolean b;      // 1 bit of storage
```

How many bits of storage does each variable require?

Question 14 [5 marks]

What does it mean if a Java *method* is declared as `static`? What are the restrictions on `static` methods? (30 words or less)

A static method is a method that can be called without having to instantiate an object (i.e., create an instance) of the class to which it belongs. Static methods are called by specifying the class name followed by the method name (e.g. `Math.sqrt()`, `StdIn.getInt()`). Without a reference to some specific object, static methods cannot access instance variables or call instance methods (i.e., variables or methods declared in a class that are not static).

Question 15 [5 marks]

What is the difference between syntax and semantics of a programming language? (20 words or less) What type of errors will a compiler, such as `Javac`, always identify for you?

The syntax of a programming language is the grammar of the language, a set of rules governing the words and symbols you can use in your program and how they are arranged.

Semantics of a programming language describes what a statement in the language does, giving the statement meaning. For example, the semantics tell us that the statement `a = b;` means the value of `b` is copied into `a`.

A statement like `a = b + c;` has correct syntax, but there is a semantic error if `a` is an `int` and `b` or `c` are `double`. The semantics define what the `+` operator means if `b` and `c` are of type `String` versus what it means if they are of type `int`.

The semantics for a language must be precise and unambiguous; every line of code in your program must have well-defined, predictable behaviour.

A compiler will *always* identify syntax errors in your program. However, it is *incorrect* to state that a compiler will *only* identify syntax errors, since it can identify semantic errors as well.

Question 16 [5 marks]

Complete the following Java program by writing a method `isLeapYear` that calculates whether a given year is a leap year (i.e., has 366 days) or not. All years that are evenly divisible by 4 are leap years, except the century years after 1582 that are evenly divisible by 100. However, of the century years after 1582, those evenly divisible by 400 are leap years. For example, 1900 is not a leap year but 1200, 1500, and 2000 are. The method should take an `int` parameter, called `year`, and return a `boolean` result.

```
class W {
    public static boolean isLeapYear(int year)
    {
        if( year % 4 != 0 )
            return false;
            // yrs NOR divisible by 4 are not LY
        if( year <= 1582 )
            return true;
            // yrs div. by 4, before (including) 1582, are LY
        if( year % 400 == 0 )
            return true;
            // yrs div. by 400, after 1582, are LY
        if( year % 100 == 0 )
            return false;
            // century years NOR div. by 400, after 1582, are not LY
        return true;
            // other div.-by-4 years are LY
    }
    public static void main (String[] args)
    {
        int y;
        while(true){
            System.out.println("Enter the year: ");
            y = StdIn.getInt();
            if( y < 0 ) break;
            System.out.print ("Is it a leap year? ");
            System.out.println( isLeapYear(y) );
        }
    }
}
```

Question 17 [5 marks]

Write a Java program that prompts for an integer n and prints out a right triangle pattern of "*" characters, with the two perpendicular sides of length n . An example of the execution of the program is shown below. Be sure to print your triangle in the *same orientation* as shown. The input typed by the user is shown in **bold**.

Enter an integer: **4**

```
*
**
***
****
```

```
class X {
    public static void main (String[] args) {
        System.out.print( "Enter an integer: " );
        int n = StdIn.getInt();

        // iterate over the number of rows
        int row;
        for (row = 0; row < n; row++){

            // print the spaces for this row
            int column;
            for (column = 0; column < (n - row - 1); column++)
                System.out.print(" ");

            // print the "*"s for this row
            // we can assert: column == (n - row - 1) here
            for ( ; column < n; column++)
                System.out.print("*");

            // end the line to begin the next row
            System.out.println();
        }
    }
}
```

Question 18 [5 marks]

Write two Java methods, `Y.max(a, b)` and `Y.min(a, b)` which accept double parameters and return a double result corresponding to the maximum or minimum of a and b . *Do not* use any if statements, rather use `Math.abs()` and properties of the formula $|a - b| + a + b$ to compute your answer. *Do not write a main() method.*

```
class Y {
    public static double max(double a, double b) {
        return ( Math.abs(a-b) + a + b ) / 2;
    }

    public static double min(double a, double b) {
        return ( a + b - Math.abs(a-b) ) / 2;
        //alternatively:
        // return ( a + b - max(a,b) );
    }
}
```

Question 19 [10 marks]

Complete the following Java program *using objects*. The goal is to remember the number entered only if it has a larger fractional part than any other number entered so far. The testing program, and example output from the program, are shown below. The input typed by the user is shown in **bold**.

Write your class BigFraction on the next page. Inside the class, write any instance variables, a constructor, and the instance method, saveBiggerFraction(BigFraction), which remembers the *num-ber* with the larger fractional and prints it out.

```

public class BigFractionTest
{
    // this is the test program for class BigFraction
    public static void main (String[] args)
    {
        BigFraction big = new BigFraction( 0.0 );
        while( true ) {

            // get a double number from the user
            System.out.print("Enter a double number: ");
            double userInput = StdIn.getDouble();
            if (userInput == 0.0 )
                break;

            // remember the number with the bigger fraction
            // don't forget to print it inside the method
            BigFraction other = new BigFraction( userInput );
            big.saveBiggerFraction( other );
        }
        System.out.println("Done.");
    }
}

Enter a double number: 5
0.0
Enter a double number: 3.25
3.25
Enter a double number: 1.3
1.3
Enter a double number: 5.1
1.3
Enter a double number: 3.3
1.3
Enter a double number: -1.7
-1.7
Enter a double number: 0
Done.

```

11

```

public class BigFraction {

    // remember the number with the largest fractional part
    private double biggest;

    public BigFraction(double initVal) {
        biggest = initVal;
    }

    // compute just the fractional part.
    // be careful about negative numbers...take Math.abs()
    private double calcFractPart() {
        // compute size of fractional part:
        // |x| - floor(|x|)
        double absX = Math.abs( biggest );
        return absX - Math.floor( absX );
    }

    public void saveBiggerFraction(BigFraction other) {
        final double myFractPart = calcFractPart();
        final double otherFractPart = other.calcFractPart();

        if( otherFractPart > myFractPart )
            biggest = other.biggest;

        // System.out.print( "The biggest fractional part so far is " );
        System.out.println( biggest );
    }
}

```

12