ICS 4U0 Semester 2, 2009 - 2010

Grade 12 Computer and Information Science
Mid-Unit Quest – Introduction to Java and Review of Programming

Mr. N. Nolfi

Victim: _____

KU	APP	TIPS	COM
/23	/10	/10	/7

1. Match each term in the left column with the ***best*** definition in the right column. ~~(17 KU)~~ (14 KU)

f class	a. A body of students who are taught together
K object	b. A tangible and visible entity
z braces	c. An appliance that corrects dental irregularities
V primitive data type	d. A method of conveying information used by "cave men"
t repetition	e. An organizational unit of Java consisting of classes and interfaces
q %	f. A template or blueprint for creating an object (it consists of methods and data fields)
l method	g. A method or data field that can be used without creating an object
r ~~index~~ doesn't count	h. A Java ***keyword*** used to create an object
j setText	i. A function
u data field	j. A method used to change the text data field of certain types of objects
aa selection	k. A concrete instance of a class
s assignment statement	l. Something students hate to get from their teachers
o getText	m. A percentage symbol
h new	n. The state of a program that is executing
y ~~instance~~ doesn't count	o. A method used to retrieve the text data field of certain types of objects
x !	p. Exclamation point
	q. Operator used to evaluate the remainder obtained upon dividing two integers
	r. A number that is used to identify a particular element of an array.
	s. A statement that is used to give a value to a variable
	t. A programming structure that allows a particular group of statements to be repeated a certain number of times or while a certain condition is true
	u. A variable declared at the global level of a class (known as a *property* in VB)
	v. A data type that is not defined in terms of simpler types
	w. Any time during which a program's source code is being edited.
	x. Java's "not" operator
	y. A method or data field that "belongs" to a particular object.
	z. Symbols used to enclose a group of statements that are to be treated as one statement
	aa. A programming structure that allows a particular group of statements to be executed while other groups of statements are ignored

2. Identify the indicated parts in each Java code snippet. **(8 KU)**

(a) `String name=new String("Bobby");`

- Name of: Class
- Name of: object
- Call of: constructor / method

(b) `int numVowels=0;`

- Name of: primitive data type
- Name of: variable
- initial value of: variable

(c) `triangleArea = base*height/2`

- assignment statement
- expression

3. The code given below is supposed to find the average of all integers between 0 and 1000 that are divisible by 5. Unfortunately, the code was written by a very sloppy programmer who made several syntax, logic and indentation errors. Circle each error and then write corrected code in the provided space. **(10 APP)**

Sloppy Code (Circle all errors)

```
sum = 0
count = 0
for (x=0, x >= 1000, ++x);{
if (x % 5 =0);
sum=+x
count++
}
average = sum/count;
```

Corrected Code

```
sum=0;
count=0;
for (x=0; x<=1000; ++x){
    if (x%5==0){
        sum+=x;
        count++;
    }
}
average = sum/count;
```

12 errors → full marks for finding 10

4. Two integers are *relatively prime* if their only common divisor is *one*. For example, 9 and 14 are relatively prime because their only common divisor is 1. However, 8 and 14 are *not* relatively prime because 2 divides into both of them.

(a) Explain the steps that a *computer* would need to execute to determine whether the integers 12 and 35 are relatively prime. **(3 TIPS, 2 COM)**

	Remainder for 12	Remainder for 35
Divide both 12 and 35 by 2 →	0	1
" " " " " " 3 →	0	2
" " " " " " 4 →	0	3
" " " " " " 5 →	2	0
⋮ continue	⋮	⋮
Divide both 12 and 35 by 12 →	0	11

At no point in this process will it be found that the remainder is zero for both cases. Therefore 12 and 35 have no common factors other than 1. They are relatively prime.

(b) Write a Java program segment that determines whether two given integers are relatively prime. Your code should include variable declarations. **(7 TIPS, 5 COM)**

```
int num1 = Integer.parseInt(editNum1.getText());
int num2 = Integer.parseInt(editNum2.getText());
int smaller, i = 2;
boolean relativelyPrime = true;

if (num1 < num2)
   smaller = num1;
else
   smaller = num2;

while (i <= smaller && relativelyPrime == true){ //begin while
   if (num1 % i == 0 && num2 % i == 0)
      relativelyPrime = false;

} //end of while

if (relativelyPrime == true)
   labelResult.setText ("Relatively prime");
else
   labelResult.setText ("Not relatively prime");
```

Communication marks alloted for

- proper indentation
- descriptive/meaningful identifier names