**CS 1301**

**Exam 4 (40pts)**

**Name:**

Multiple Choice: (4 x 2 = 8pts)

1. Suppose your method does not return any value, which of the following keywords can be used as a return type?

a. void

b. int

c. double

d. public

2. Arguments to methods always appear within \_\_\_\_\_\_\_\_

a. brackets

b. parentheses

c. curly braces

d. quotation marks

3. Given the following method

public static void nPrint(String message, int n){

while(n > o){

System.out.print(message);

n--;

}

}

What is the printout of the call nPrint('a', 4)?

a. aaaaa

b. aaaa

c. aaa

d. invalid call

4. Having multiple class methods of the same name where each method has a different number of or type of parameters is known as

a. encapsulation

b. information hiding

c. tokenizing

d. importing

e. method overloading

1. What output is produced by the following code fragment if lines = 6? (6pts)

public void nStars (int lines){

for (int i = 1; i <= lines; i++){

System.out.println();

for (int s = 1; s < i; s++)

System.out.print ("$");

for (int d = 1; d <= lines -i + 1; d++)

System.out.print("\*");

}

}

>

>\*\*\*\*\*\*

>$\*\*\*\*\*

>$$\*\*\*\*

>$$$\*\*\*

>$$$$\*\*

>$$$$$\*

6. State true or false: (6 x 1.5 = 9pts)

True a) A return statement is required for a value-returning method.

False b) The value of the parameter is passed to the argument.

True c) Ambiguous invocation causes a compilation error.

False d) One cannot declare a local variable with the same name multiple times in different non-nesting blocks in a method.

True e) The scope of a parameter covers the entire method.

True f) The scope of a local variable starts from its declaration and continues to the end of the block that contains the variable.

1. Write a method (not entire program) that accepts three integer parameters and returns the median of the three integers. You can assume all the integers are distinct.

For example, if the values are 7, 5, 6 the method will return 6. If the values are 2, 3, 1 the method will return 2. (7pts)

public class Q6

{

/\* Check the median number from three integers

\* @param a integer number

\* @param b integer number

\* @param b integer number

\* @return the median integer number

\*/

public static int getMax(int a, int b, int c)

{

if(a < b && a > c) return a;

if(a < c && a > b) return a;

if(b > a && b < c) return b;

if(b > c && b < a) return b;

return c;

}

public static void main(String args[])

{

System.out.println(getMax(7, 5, 6));

}

}

1. Write a program, read two integers from the keyboard. Finds whether the two numbers are relatively prime to each other or not. Two integers are **relatively prime** if there is no integer greater than one that divides them both. (10pts)

Extra credits: write doc string for functions (2pts).

import java.util.Scanner;

public class Q8

{

/\* Calculate the Greatest Common Divisor

\* @param a integer number

\* @param b integer number

\* @return gcd

\*/

public static int gcd(int a, int b)

{

int gcd = 1;

int k = 1;

while(k <= a && k <= b)

{

if (a%k == 0 && b%k == 0)

gcd = k;

k++;

}

return gcd;

}

/\* Check if two integers are relative primes, or so called coprime integers

\* @param a integer number

\* @param b integer number

\* @return true, if two integer are relatively prime; false, otherwise

\*/

public static boolean coPrimes(int a, int b)

{

if(gcd(a, b) == 1)

return true;

else

return false;

}

public static void main(String [] args)

{

Scanner input = new Scanner(System.in);

System.out.println("Enter two integer numbers:");

int a = input.nextInt();

int b = input.nextInt();

if(coPrimes(a, b))

System.out.printf("%d and %d are relatively prime \n", a, b);

else

System.out.printf("%d and %d are not relatively prime \n", a, b);

}

}