COMP 2243 Test 2 Review

Part A: What results are printed by each of the following blocks of code? (20± points)

6 to 8 questions related to *for loop, do…while loop, nested loop, method*

Part B: Methods (20± points)

1. Method overloading concept (5± points)
2. Write Java code to call methods in a procedural program. (6± points)
3. Write Java code to call methods in an OOP program. (9± points)

Part C: Coding (60± points)

1. (8± points) **do..while** or **for** loop
2. (10± points) **for** or **do..while** loop
3. Procedural programming with methods (12± points)

   Write method definitions

   Call methods
4. Programming with File I/O (12± points)

   Read data from a data file

   Calculate

   Write the result to an output file
5. OOP (18± points)

   Class definition

   Create objects

   Print objects

   Call object's methods

   Compare objects

   Change objects

Sample questions:

Part A What results are printed by each of the following blocks of code?
4 questions related to **do..while**, **for**, and **nested** loops, about 12± points

`int i = 100;`
 	
`do {`
 	
`  System.out.print(i + " ");`
 	
`  i++;`
 	
`} while (i <= 110);`
	`int i = 100;`
	
	`for( int i = 100; i<= 110; i++`
	
	`  System.out.print(i + " ");`
	
	`}`

`int i = 110;`
 	
`do {`
 	
`  System.out.print(i + " ");`
 	
`  i--;`
 	
`} while (i >= 100);`
	`for (int i = 1; i <= 5; i++) {`
	
	`  for (int x = 1; x <= i; x++) {`
	
	`  System.out.print("M");`
	
	`  }`
	`  System.out.println();`
	`}`

`int i = 5, x = 100;`
 	
`do {`
 	
`  x++;`
 	
`  i--;`
 	
`} while( i >= 0  );`
 	
`System.out.println( x );`
	`int x = 100;`
	
	`for (int i = 10; i > 5; i--) {`
	
	`  x++;`
	
	`}`
	
	`System.out.println( x );`

`int i = 0,  s = 0;`
 	
`do {`
 	
`  s += i;`
 	
`  i += 5;`
 	
`} while( i <= 10 )`
 	
`System.out.println(s);`
	`int i = 0,  s = 0;`
	
	`for( int i = 0; i <= 10; i += 5;`
	
	`  s += i;`
	
	`}`
	
	`System.out.println(s);`

```
int i = 10, x = 50;

do {

    if( i%5 == 0 ) {

        System.out.print(i + " ");

        x += 10;
    }

     i++;
} while( i <= 15 );

System.out.println(x);
```

```
int n = 2;

do {
      System.out.print(n + " ");
      n++;
} while (n >= 6);


System.out.println();
```

What results are printed by each of the following blocks of code?
2 questions related to **methods**, about 8± points

```
public static void main (String [] args) {

     int n1 = 10, n2 = 20, x;

     x = method1 (n1, n2);

     System.out.println( x );

     System.out.println( method2(x) );

     System.out.println ( method3( n2) );
}

public static double method1 (int a, int b) {
       //if else or loop to change a and b

       return a + b;
}

public static boolean method2 (int n){
    //if else or loop to change n

    if (n > 0) {
       return true;
    }
    else {
       return false;
}

public static String method3 (int n){
    //if else or loop to change n

     return "Hello " + n;
}
```

Part B: Methods (20± points)

1. Write Java code to call methods in procedural program.

```
import java.util.Scanner;

public class PayrollWithDecisionMethod {

   static final double REGULAR_HOURS = 40;
   static final double OVERTIME_RATE = 1.5;

   public static void main(String [] args) { //entry of the program

      Scanner keyboard = new Scanner(System.in);

      String firstName, lastName, query;
      double hoursWorked, baseWage, weeklyPay;

      System.out.print("Enter first name: ");
      firstName = keyboard.next();   //it stops at a blank space or a return

      System.out.print("Enter last name: ");
      lastName = keyboard.next();

      System.out.print("Enter hours worked: ");
      hoursWorked = keyboard.nextDouble();

      System.out.print("Enter base wage: ");
      baseWage = keyboard.nextDouble();


          //call calculatePayroll() method and pass arguments


      ____________________________________________________________

          //call printPayroll() method


      ____________________________________________________________

   } //end main

   public static double calculatePayroll(double hours, double base)  {

      return hours * base;
   }

   public static void printPayroll(String first, String last,
                                   double hours, double base,
                                   double pay) {

      System.out.println("\n" +
                         first + " " + last + " Weekly Payroll Report: \n" +
                         "Hours Worked: " + hours + "\n" +
                         "Base Wage: $" + base + "\n" +
                         "Weekly Pay Before Tax: $" + String.format("%.2f", pay));

   }

}  //end class
```

2. Is there any problem with the following method overloading? Why?

```
public static double computeArea( double x ) {

    return (Math.PI * x * x);
}

public static double computeArea( double x, double y ) {

    return ( x * y );
}

public static double computeArea( double n ) {

    return ( n * n );
}
```

3. Write Java code to call methods in OOP program.

```
public class Car {                                          //Car.java
     private String make;
     private int year;

     public Car(String m, int y) {
         make = m;
         year = y;
     }

     public String description(String message, int n){
         return make + " " + message + " " + n;
     }

     public static void Hello(String n)  {
         System.out.println("Welcome to " + n);
     }
 }

*****************************************************************************
public class CarClient {                                    //CarClient.java
      public static void main(String [] args ) {

           String s1 = "Ford", s2 = "F-159", s3 = "Ford Dealer";

           int y = 2022;

           Car myCar;


           _____________________________________________    /*Create a Car object, then
                                                              call description method and
                                                              print the returned value*/


           _____________________________________________


           myCar = null;    //myCar object is deleted.


           _____________________________________________    //Call Hello() method


   }
}
```

Part C: Coding (60± points)

1. (8± points) Rewrite the loops from **do..while** to **for**, **for** to **do..while**

2. (10± points) Write a **for** loop or **do .. while** to calculate a sequence of numbers, sum, and average.

3. (12± points) Given a complete program, modify it with methods.

4. (12± points) Programming with File I/O

5. (18± points) OOP