Practice Exam 1 - COP 3502 Spring 2022

Module 0: Introduction to Computer Science

Q1. What does the Java Compiler produce as output?

- Byte Code

Q2. What is the input provided to the Java Compiler?

- Source Code

Q3. How can you comment multiple lines in Java?

- /* */

Module 1: Variables and Arithmetic

Q1. What would be printed by the following code?

```
public static void main(String args[])
  {
     int a = 12, b = 2;
     int c = a / ++b;
     int d = b / a;
     System.out.println(c + " " + d);
  }
```

- 4 0

Q2. What would be printed by the following code?

```
public static void main(String[] args)
 {
     int a = 5, b = 2;
     a *= 5;
     a += b++;
     System.out.print(a++);
     System.out.print(++b);
     System.out.print(a++);
 }
```

- 27428

Q3. What would be printed by the following code?

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

```
        int a = 5, b = 2;
        b = a > b ? a++ : ++a;
        System.out.print(b);
    }
```

- 5

Module 2: Program Control

Q1. Predict program output:

```
public static void main(String[] args){
      int i = 5;
      switch(3%i)
      {
          case 0:
            System.out.print("1");
          case 1:
            System.out.print("2");
          case 2:
            System.out.print("3");
          case 3:
            System.out.print("4");
          case 4:
            System.out.print("5");
      }
}
```

- 45

Q2. What is the output of the following program?

```
int nyans = 20;
if (nyans < 20 && 5 < 12 || nyans < 30)
    System.out.println("True");
else
   System.out.println("False");
```

- True

Q3. What is the output of the following program?

```
public static void main(String[] args) {
    int a = 3;
    int b = 4;
    int c = 6;

    if (b < 2 || b > 3) {
        a -= 2;
    }
    if (c > 4) {
        a -= 1;
    }
    System.out.println(a);
}
```

- 0

Q4. What is the output of the following program

```
public static void main(String[] args) {
    for (int acc = 2; acc < 20; acc += 3) {
        System.out.print(acc + " ");
        if (acc % 5 == 0 || acc % 4 == 0) {
            break;
        }
        if (acc % 3 == 0 || acc % 2 == 0) {
            acc++;
            continue;
        }
    }
}
```

- 2 6 10

Q5. Find Errors in the following program assuming its goal is to print out all the divisors of numbers in the range [i, sum].

```
int sum = 80;
int i = 0;

//i was already declared
for (int i = 1; i <= sum; i++)
{
     if (i % 2 = 0) { //should use ==
          System.out.println(2 + " is a divisor of " + i);
     }
```

```
        //should use if
        else if (i % 3 = 0){
             System.out.println(3 + " is a divisor of " + i)  //semicolon
        }
        //should use if
        else if (i % 5 = 0)  {
             System.out.println(5 + "is a divisor of " + i)  //semicolon
        }
    }
```

Q6. Find the errors in the following code snippet.

```
        public static mysterious(x) {
            double x = 2; j;
            double y = 0;
            for(;i <= num;) {
                y += x;
            }
        }
```

1) mysterious method doesn't not have a return type;
2) x as a parameter doesn't have a data type;
3) when j is declared, it should be separated using "," if j is of type double.
4) i is not initialized and then used for comparison with the num value;
5) num is not declared and initialized;
6) i is not updated in the loop; there is no way to exit the loop. We will have infinite loop.

Module 3: Methods and Number System

Q1. Predict the output of the following code.

```
public static void main(String[] args)
{
   int num1 = 2;
   int num2 = 8;
   int num3 = 6;
   num1 = multi(num2, num3);
   System.out.println(num1);
}

public static int multi(int var1, int var2)
{
   int answer = var2 + var2 * var1;
   return answer;
}
```

- 54

Q2. Predict the output of the following code.

```
public static void main(String[] args)
{
   int num1 = 4;
   int num2 = 9;
   int num3 = 18;
   multi(num2, num3);
   System.out.println(num3);
}

public static int multi(int num2, int num4)
{
   int num3 = num2 + num4 / num2;
   return num3;
}
```

- 18

Q3. Convert the binary number 0b01110011 to decimal number.

- 115

Q4. Convert the octal number 77 to a binary number.

- 111111

Q5. Covert the decimal number 88 to binary, octal and hexadecimal number.

- 1011000_2
- 130_8
- 58_{16}

Module 4A: Data Types

Q1. Predict the output of the following program.

```
public static void main(String[] args) {
      double num1 = 17/2;
      double num2 = (double) (17/2);
      double num3 = (double) 17 / 2;
      double num4 = (double) 17 / (double) 2;
      System.out.println(num1);
```

```
        System.out.println(num2);
        System.out.println(num3);
        System.out.println(num4);
    }
```

- 8.0
- 8.0
- 8.5
- 8.5

Q2. What is the output of the following program?

```
public class MyClass {
 enum Level {
   FIRST,
   SECOND,
   THIRD
 }

 public static void main(String[] args) {
   Level myVar = Level.SECOND;
   System.out.println(myVar);
 }
}
```

- SECOND

Q3. What is the output of the following program?

```
public static void main(String[] args) {
    char c1 = 'A';
    char c2 = 'c';
    char c3 = 'e';

    if (c1 == Character.toUpperCase('a')) {
        System.out.print("Is it 1?");
    }
    if (c2 > c3) {
        System.out.println(" No.");
    }
    else {
        System.out.println(" Or 11?");
    }
}
```

- Is it 1? Or 11?

Q4. What is the output of the following program?

```
public static void main(String[] args) {
      String name = "Al E. Gator";
      String sName = "";


      for (int i = 0; i < name.length(); i++) {
          if (name.charAt(i) == ' ') {
              continue;
          }
          if (name.charAt(i) == '.') {
              break;
          }
          sName = sName + name.charAt(i);
      }
      System.out.println(sName);
  }
```

- AlE

Q5. Given the following code, predict the value of variable var1 and var2.

```
public static void main(String[] args) {
        int num1 = 3;
        int num2 = 8;
        boolean var1 = !(++num1 >= 3);
        int var2 = var1 ? ++num2 - num1-- : num1++ + --num2;
}
```

- var1 = false
- var2 = 11

Coding

Q1. Implement your own version of the built-in substring method of the Java library. The function header for the substring method is as follows:

```
public String substring(String s, int begIndex, int endIndex)
{

}
```

This method **returns** a **new string** that is a substring of the string, s. The substring begins with the character at the specified index and **extends** up **to endIndex – 1**

Note: Only write the method and any helper methods, no need to create a class. You are not allowed to use *substring( )* method from the Java library, but feel free to use functions such as *charAt( )* and *length( )*.

```
String result = "";
for (int i = begIndex; i < endIndex; i++)
   result += s.charAt(i);
return result;
```

Q2. Write a method *identicalDigits(int num)* that takes in an integer num in the range of 10 – 90. This method doesn't return anything. It prints out a countup starting from the integer num, and stopping when both output digits are identical. You must use a loop.

Example 1

num = 18

Output: 18 19 20 21 22

Example 2

num = 66

Output: 66

Note: For coding simplicity, follow each output number by a space, even the last one. You will assume num value is always in the range of 10 – 90. There is no invalid num value passed in.

```
public static void identicalDigits(int num) {
   for (int i = num; i < 100; i++) {
      System.out.print(i + " ");
      if (i / 10 == i % 10)
         break;
   }
}
```

Q3. Write a method *printTriangle(int base)* that takes in a positive integer base and prints a triangle made of asterisks with a base of the given size.

Example 1: base = 5 Output: * ** *** **** *****	Example 2: base = 7 Output: * ** *** **** ***** ******

	*******

```
public static void printTriangle(int base) {
   for (int i = 1; i <= base; i++) {
      for (int j = 0; j < i; j++) {
         System.out.print("*");
      }
      System.out.println();
   }
}
```

BONUS: Write a method *printInverseTriangle(int base)* that prints the same triangle pattern from above, but with the base at the top.

Example 1: base = 5 Output: ***** **** *** ** *	Example 2: base = 7 Output: ******* ****** ***** **** *** ** *

```
public static void printInverseTriangle(int base) {
   for (int i = base; i >= 1; i--) {
      for (int j = 0; j < i; j++) {
         System.out.print("*");
      }
      System.out.println();
   }
}
```

Q4. A prime number is a whole number greater than 1, which is only divisible by 1 and itself. Write a method to detect whether a number is prime or not. The function header is as follows:

```
public static boolean isPrime(int n){
        // Corner case

        if (n <= 1)

            return false;

        // Check from 2 to n-1
```

```
        for (int i = 2; i < n; i++)
            if (n % i == 0)
                return false;
        return true;
}
```

Q5. A user will enter an initial number, followed by that number of integers. Output those integer's sum. Repeat until the initial number is 0 or negative.

Ex 1: if the user enters 3 9 6 1 0, the output is 16.

- Explanation: 3 is the initial number that represents you will add up the following 3 integers 9 + 6 + 1 = 16. Stop when you encounter 0 as the initial value.

Ex 2: if the user enters 3 9 6 1 2 5 3 0, the output is

16

8

- Explanation: 3 is the initial number that represents you will add up the following 3 integers 9 + 6 + 1 = 16. Then 2 is the initial number that represents you will add up the following 2 integers 5 + 3 = 8. Stop when you encounter 0 as the initial value.

```
public static void main(String[] args) {
      Scanner scnr = new Scanner(System.in);
      int numInts = 0;
      int intsSum = 0;
      int userInt = 0;
      int i;

      numInts = scnr.nextInt();

      while (numInts > 0) {

         intsSum = 0;
         for (i = 0; i < numInts; ++i) {
            userInt = scnr.nextInt();
            intsSum += userInt;
         }
         System.out.println(intsSum);

         numInts = scnr.nextInt();
      }
   }
```

Q6. Write a method indexOfString(String str1, String str2) that returns the index of the first occurrence
of the specified str2 in str1. If it does not occur as a substring, -1 is returned.

Example: str1 = “Good Morning”, str2 = “od”, return 2

Example: str1 = ”Good Morning”, str2 = “op”, return -1

```
public static int indexOfString(String str1, String str2) {
      int len1 = str1.length();
      int len2 = str2.length();

      for (int i = 0; i < len1; i++) {
          if (i + len2 <= len1) {
              String sub = str1.substring(i, i + len2);
              if (sub.equals(str2)) {
                  return i;
              }
          }
      }
      return -1;
}
```