SERIAL

University of Bahrain
College of Information Technology
Department of Computer Science
Second Semester, 2018-2019
ITCS 113 / ITCS 111
Computer Programming I

A

FINAL EXAM

Date: 13th June 2019

Duration: 2 Hours

STUDENT NAME	

STUDENT ID #								

SECTION #	

□ **ITCS111**
□ **ITCS113**

NOTE: THERE ARE (7) PAGES IN THIS TEST
WRITE ONLY ONE SOLUTION FOR EACH QUESTION

Question #	MARKS	SCORES
1	10	
2	13	
3	13	
4	24	
TOTAL	60	

Question 1 (10 Points) – Form A

Please choose the best correct answer from the given choices. Please choose only one.

1. Which of the following statement prints `smith\exam1\test.txt` ?

```
(a) System.out.println("smith\exam1\test.txt");
(b) System.out.println("smith\\exam1\\test.txt");
(c) System.out.println("smith\"exam1\"test.txt");
(d) System.out.println("smith"\exam1"\test.txt");
```

2. Suppose you have the string s="SELECT". What is the return value of s.substring(0, 5)?

```
(a) "SELE"
(b) "ELECT"
(c) "SELECT"
(d) "SELEC"
```

3. Consider the code on the right⇨

Which of the following is true?

(a) If **a** is true and **b** is true then the output is "A && B"

(b) If **a** is true and **b** is false then the output is "notB"

(c) If **a** is false and **b** is true then the output is "ELSE"

(d) If **a** is false and **b** is false then the output is "ELSE"

```
// a and b are Boolean variables
if( a )
    System.out.println("A");
else if(a && b)
    System.out.println( "A && B");
else {
   if ( !b )
      System.out.println( "notB") ;
   else
      System.out.println( "ELSE" ) ;
 }
```

4. Consider the code on the right⇨

Which data types are acceptable types for **x**?

(a) **int** and **float**

(b) **char** and **double**

(c) **int, short,** and **String**

(d) **long, float** and **double**

```
switch(x) {
    default:
        System.out.println("Hello");
}
```

5. A constructor

(a) always accepts two arguments

(b) has the same name as the class

(c) has return type of void

(d) always has an access specifier of private

6. For the following code, which statement is true?

```
public class Circle {
      private double radius;
      public double x;
      private double y;
}
```

(a) x is not available to code that is written outside the `Circle` class.

(b) `radius` is available to code written outside the `Circle` class.

(c) `radius`, x, and y are methods of the `Circle` class.

(d) y is not available to code that is written outside the `Circle` class.

7. How many times will the following `do-while` loop be executed?

```
int x = 11;
do
{
      x += 20;
} while (x > 100);
```

(a) 0

(b) 1

(c) 4

(d) 5

8. Which of the following is the header definition of a method called `test()`, if it has three integer input parameters and returns a boolean data type.

(a) `public boolean test(int x1, x2, x3)`

(b) `public test(int x1, x2, x3):boolean`

(c) `public boolean test(int x1, int x2, int x3)`

(d) `public boolean test(int x1, int x2, int x3):void`

9. Suppose **num** is a private integer. Which of the following is its mutator (set) method?

```
(a) public void setNum(int num) {
        System.out.println(num);
    }
```

```
(b) public void setNum(int num){
       this.num = num;
    }
```

```
(c) public int setNum(int num) {
       return num;
    }
```

```
(d) public int setNum(int num) {
        return this.num + num;
    }
```

10. Which of the following loops print the positive numbers (including 0) that are less than 100 and are divisible by 5?

```
(a) for(int i=1; i < 100; i++)
      if (i / 5 == 0)
        System.out.println(i);
```

```
(b) for(int i=1; i < 100; i+=4)
        System.out.println(i);
```

```
(c) int i = 0;
    do {
       i += 4;
       System.out.println(i+1);
    } while (i < 100);
```

```
(d) for(int i=0; i < 100; i+=5)
        System.out.println(i);
```

Question 2 (13 Points)

Write a Java program to sell a limited number of cinema tickets. Each person can buy up to 4 tickets, where the number of available tickets is 100. Your program should do the following:

1. Prompt the user for the number of tickets to buy.
2. Display the number of remaining tickets.
3. Repeat the above steps until all tickets have been sold.
4. Display the total number of buyers.

Invalid testing: (1) If the number of tickets to buy is negative or zero, then print an invalid message. (2) If the number of tickets to buy is more than 4, then print a message that the maximum tickets to buy is 4. (3) If the number of tickets to buy is more than the available tickets, then print a message that there are no enough tickets available. See sample input/output on the right side ⇨

SAMPLE INPUT/OUTPUT

```
How many tickets do you want to buy? 3
Available tickets = 97
How many tickets do you want to buy? 6
Sorry. Maximum tickets to buy is 4.
…..
How many tickets do you want to buy? 4
Available tickets = 2
How many tickets do you want to buy? 3
Sorry. Only 2 tickets are left.
How many tickets do you want to buy? 2
Total number of buyers is 32
```

```java
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        int remainingTickets = 100 , buyersCount = 0;

        Scanner sc = new Scanner(System.in);
        int tickets;
        while(remainingTickets>0){
            System.out.print("How many tickets do you want to buy? ");
            tickets = sc.nextInt();
            if (tickets<1) {
                System.out.println("Invalid! Minimum tickets to buy is 1.");
            }
            else if(tickets > 4) {
                System.out.println("Sorry. Maximum tickets to buy is 4.");
            }
            else if(tickets > remainingTickets) {
                System.out.println("Sorry. Only "+ remainingTickets + “ tickets are 
left.”);
            }
            else {
            buyersCount++;
            remainingTickets -= tickets;
            System.out.println(“Available tickets = “ + remainingTickets); }
        }

        System.out.println(“Total number of buyers is “ + buyersCount);
    }

  }
```

Question 3 (13 Points)

Write a java program that uses nested **for-loops** to display a sequence of numbers. The program should ask the user to enter an integer n. The number n should be in the range 3 to 1000. If the number is not in the range, then display an invalid message and exit the program. Otherwise, on the first line, the program should display the number n repeated n times. Then on the next line it should display the number $n-1$ repeated $n-1$ times, and on the next line it should display the number $n-2$ repeated $n-2$ times, and so on for the next lines and stop when the number reaches zero. Also, display a star (*) before each line and display a hash (#) at the end of each line.
See sample input/output on the right side ➪

SAMPLE INPUT/OUTPUT

If $n = 2$	If $n = 3$	If $n = 6$
Invalid number	*333# *22# *1#	*666666# *55555# *4444# *333# *22# *1#

```java
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        System.out.println("enter an integer n between 3 and 1000:  ");
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();

        if (n<3 || n>1000) {
            System.out.println("Invalid number");
            System.exit(0);
        }

        for (int i = n; i > 0; i--) {
            System.out.print("*");
            for (int j = 0; j < i; j++) {
                System.out.print(i);
            }
            System.out.println("#");
        }

    }

  }
```

Question 4 (17+7 Points)

Part (1) Define a class with the following specification:

a. The class name is **Exam** with three private data members: exam course **code** (String), exam **score** (double) for the student, and maximum exam score **maxScore** (double).

b. Write a constructor that initializes all the data members of the class from its parameters. If **maxScore** is less than **score**, then set them to zero and print a message for this change.

c. Provide an accessor (get) method and a mutator (set) method *only* for the **score** data member. Make sure that the new score in the set method does not exceed **maxScore**, otherwise leave the current value of **score** unchanged.

d. Define a method called **addBonus()** that adds a given bonus score, as input parameter, to the student's score. If the new score after adding the bonus exceeded the maximum score, then set **score** to **maxScore.**

e. Define a method called **percentage()** that returns the exam percentage out of 100.

f. Define a method called **displayExamDetails()** to display on the screen exam details as in the following sample:

```
Code:          ITCS113
Score:         53.5
Max Score:     60.0
Percentage:    89.167%
```

```
public class Exam{

    private String code;
    private double score;
    private double maxScore;

    Exam(String code,double score,double maxScore){
        this.code = code;
        if(score>maxScore){
            System.out.println("Score cannot be greater than max score. Initializing both
to 0");
            this.score=0;
            this.maxScore=0;
        }
        else{
            this.score = score;
            this.maxScore = maxScore;
        }
    }

    public void setScore(double score) {
        if(score <= this.maxScore){ this.score = score; }
    }

    public double getScore() {
        return score;
    }

    public void addBonus(double bonus){
        if(this.score+bonus > this.maxScore){ this.score = this.maxScore;}
        else{this.score += bonus;}
    }

    public double percentage(){ return (this.score/this.maxScore) * 100;}

    public void displayExamDetails(){
        System.out.println("Code: " + this.code);
        System.out.println("Score: " + this.score);
        System.out.println("Max Score: " + this.maxScore);
        System.out.println("Percentage: " + this.percentage() + "%");
    }

}
```

Part (2) Write a Java application to do the following:

a. Create an object called **exam1** from the class **Exam** and initialize its course name to “ITCS113”, its score to 53.5 and its maximum score to 60.0.
b. Ask the user for the bonus score, and then add this score to the current score
c. Display the exam details.
d. Display the amount of percentage increase after adding the bonus. For example, if the current percentage is 89.167 and the percentage after adding the bonus is 94.1267, then the increase is 94.1267 - 89.167 = 5.

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Exam exam1 = new Exam("ITCS113", 53.5, 60.0);

        System.out.println("Enter bonus score: ");
        Scanner sc = new Scanner(System.in);
        double bonus = sc.nextDouble();

        double oldPercentage = exam1.percentage();
        exam1.addBonus(bonus);
        double newPercentage = exam1.percentage();

        exam1.displayExamDetails();

        System.out.println("Percentage increase is: " +
        (newPercentage - oldPercentage) );

    }

  }
```