King Saud University

College of Computer and Information Sciences
Computer Science Department

Duration	2h: 30 minutes
Course Code:	CSC 111
Course Title:	Introduction to Programming
Semester:	Spring 2015
Exercises Cover Sheet:	**Final Exam – (Version A B C)**

Student Name:
Student ID:
Student Section No.

ANSWER SHEET (ALL FORMS)

Tick the Relevant	Computer Science B.Sc. Program ABET Student Outcomes	Question No. Relevant Is Hyperlinked	Covering %
√	**a) Apply knowledge of computing and mathematics appropriate to the discipline;**	**1,2,3**	**50%**
	b) Analyze a problem, and identify and define the computing requirements appropriate to its solution		
√	**c) Design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;**	**4,5**	**50%**
	d) Function effectively on teams to accomplish a common goal;		
	e) Understanding of professional, ethical, legal, security, and social issues and responsibilities;		
	f) Communicate effectively with a range of audiences;		
	g) Analyze the local and global impact of computing on individuals, organizations and society;		
	h) Recognition of the need for, and an ability to engage in, continuing professional development;		
	i) Use current techniques, skills, and tools necessary for computing practices.		
	j) Apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices;		
	k) Apply design and development principles in the construction of software systems of varying complexity;		

Question 1 (11 Marks)

Put your answer of question 1 (**multiple choice questions**) in the following table:

FORM A B C	
Question	Answer
1	**A**
2	**C**
3	**E**
4	**C**
5	**C**
6	**D**
7	**B**
8	**A**
9	**B**
10	**B**
11	**B**

1. Suppose a Scanner object is created as follows:

```
Scanner input = new Scanner(System.in);
```

What method do you use to read an int value?

A. input.nextInt();
B. input.nextInteger();
C. input.int();
D. input.integer();

---

2. What is the exact output of the following code?

```
double area = 3.5;
System.out.print("area");
System.out.print(area);
```

A. 3.53.5
B. 3.5 3.5
C. area3.5
D. area 3.5

---

3. Which of the following is a valid identifier?

A. private
B. class
C. 9X
D. 8+9
E. radius

---

4. _____ is the Java assignment operator.

A. ==
B. :=
C. =
D. =:

---

5. The expression 4 + 20 / (3 - 1) * 2 is evaluated to

A. 4
B. 20
C. 24
D. 9
E. 25

---

6. Suppose x is 1. What is x after x += 2?

A. 0
B. 1
C. 2
D. 3
E. 4

---

7. Given the following method

```
public void powersOfTwo (){
        int power = 1;
        for (int count=1; count <= 3; count++) {
                power *= 2;
        }
        System.out.print (power);
}
```

What will be displayed by the call powersOfTwo() ?

A. 1
B. 8
C. 16
D. 4

---

8. Given the following method

```
public void isAlpha (char ch) {
        System.out.print ( (ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z') );
}
```

What will be displayed by the call isAlpha('%') ?

A. false
B. true
C. FALSE
D. Nothing because there is an error

---

9. Given the following method

```
public void average (int num1, int num2, int num3){
        System.out.print ((num1+num2+num3) / 3.0);
}
```

What will be displayed by the call average(2,3,4) ?

A. 3
B. 3.0
C. 9
D. Nothing because there is an error

---

10. Given the following method

```
public void countA (String str){
        int count = 0;
        for (int index=0; index < str.length(); index++)
                if (str.charAt(index) == 'A')
                        count++;
        System.out.print(count);
}
```

What will be displayed by the call countA("I like JAVA language") ?

A. 4
B. 2
C. 12
D. 20
E. Nothing because there is an error

---

11. Given the following method

```
public void alarm (int num){
        for (int count=1; count <= num; count++)
                System.out.println ("Alarm!");
        }
```

What will be displayed by the call alarm(2) ?

A. Alarm!

B. Alarm!
Alarm!

C. Alarm!
Alarm!
Alarm!

E. Nothing because there is an error

Question 2 (10 Marks)

The following code segments may or may not have errors (logical, syntactical, or given certain input: runtime). If there are errors, **underline their positions** and **rewrite the whole code segment after correcting all the errors**. Assume that all necessary imports and method headers are given before the code segment.

Purpose of code	Code segment (if there are are errors, underline the errors and re-write the code corrected, if there are no errors do not do anything)
1) Declare and create an array of integers and initialize it to the first 5 positive integers.	`int x = new int[];` `x[0] = 1; x[1] = 2; x[2] = 3; x[3] = 4; x[4] = 5;`

Correction:	`int x[] = new int[5];`
	`x[0] = 1; x[1] = 2; x[2] = 3; x[3] = 4;`
	`x[4] = 5;`
	
	OR
	
	`int x[] = {1,2,3,4,5}`

2) Find the index of the second smallest element in the array a	`smallest = 0;`
	`for (int 1=i; i<a.length; i+=1)`
	`  if (a[i]<a[smallest])`
	`  smallest = a;`
	`if (smallest!=0)`
	`  second_smallest = 0;`
	`else`
	`  second_smallest = 1;`
	`for (int i=second_smallest+2; i<=a.length; i++)`
	`  if (a[i]<a[second_smallest])&&(i!=smallest)`
	`  second_smallest = i;`

Purpose of code	Code segment (if there are are errors, underline the errors and re-write the code corrected, if there are no errors do not do anything)

Correction:	`smallest = 0;`
	`for (int i=1; i<a.length; i+=1)`
	`  if (a[i]<a[smallest])`
	`  smallest = i;`
	`if (smallest!=0)`
	`  second_smallest = 0;`
	`else`
	`  second_smallest = 1;`
	`for (int i=second_smallest+1; i<a.length; i++)`
	`  if (a[i]<a[second_smallest])&&(i!=smallest)`
	`  second_smallest = i;`

3) Read form the user the number of students, their GPAs, and calculate the average GPA.	`Scanner kb = new Scanner(System.in);`
	`int n_students = kb.nextInt();`
	`int sum = 0`
	`for (int i=0; i<=n; i++)`
	`  sum += kb.nextInt();`
	`int average = -1;  // the -1 means invalid value`
	`if (true)`
	`  average =/ sum;`

Correction:	`Scanner kb = new Scanner(System.in);`
	`int n_students = kb.nextInt();`
	`int sum = 0;`
	`for (int i=0; i<n; i++)`
	`  sum += kb.nextInt();`
	`double average = -1;  // the -1 means invalid value`
	`if (n_students!=0)`
	`  average =/ sum;`

Question 3 (4 Marks)

What is the output of the following java program?

```
class Football {
  private double price;
  private String brand;
  private String quality;

  public Football(double pr, String br) {
    price = pr;
    brand = br;
    quality = "Good";
    display();
  }

public void changePrice(double p) {
    price = p;
    if (price < 0){
      price = 0.0;
      System.out.println("cant bye a football");
    }
    else {
      quality = "Excellent";
      changeBrand();
      display();
    }
  }

  private void changeBrand() {
    if (price >= 0 && price <= 100.0)
      brand = "Local";
    else if (price > 100 && price <= 200.0)
            brand = "Unknown";
         else
            brand = "Adidas";
  }

  private void display(){
    System.out.println("Price of ball "+ price);
    System.out.println("Brand is "+ brand);
    System.out.println("Has " + quality+" quality");
  }
}

public class FootballTest {
  public static void main(String[] args) {
    System.out.println("First state :");
    Football fb = new Football(125.5," Unknown");
    System.out.println("After change :");
    fb.changePrice(500.0);
  }
}
```

Answer

First state :

Price of ball 125.5

Brand is Unknown

Has Good quality

After change :

Price of ball 500.0

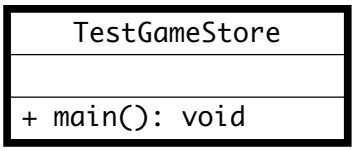
Brand is Adidas

Has Excellent quality

Question 4 (15 Marks)

Write a Java program to manage a game store. The program allows the storeowner to add new games to inventory and to sell games. Game data is stored in two parallel arrays: `name` and `price`. Use the UML of the class below.

GameStore
- name: String[] - price: double[] - numOfGames: int - totalSale: double
+ GameStore() + GameStore(inventorySize: int) + add(name: String, price: double): void + findGame(name: String): int + sell(name: String): void + getTotalSale(): double



Here is a description of each method in the class:

- `GameStore()`: this is the default constructor. It calls the other constructor and pass it the value 100 as default inventory size.
- `GameStore(inventorySize: int)`: this constructor creates the two arrays of a size based on inventorySize. It also initializes numOfGames to zero.
- `add`: this method is used to add a game to the store given its name and price. If the game is already in the store (use method `findGame` to check this) then it prints an error message "**ERROR: GAME ALREADY IN STORE**". Notice that a game is added only if there is enough space in inventory otherwise add prints an error message "**ERROR: NO ENOUGH SPACE IN INVENTORY**".
- `findGame`: this method is used to find game's data given its name (use method `equals()` to compare strings). The method returns the array index at which the game's data is stored otherwise it returns -1 indicating that the game was not found.
- `sell`: this method is used to sell a game given its name. If the game is not in the inventory then it prints an error message "**ERROR: GAME IS SOLD OUT**". If a game is sold then it must be deleted from inventory. Each time a game is sold, we add its price to `totalSale`.

Write program `TestGameStore` that does the following:

1) Starts by creating an object of type `GameStore` that can store data of up to N games (inventory size is N) where N is an integer input by user.
2) Then it adds 3 games to inventory. It reads name of the game (as a String) and its price (as a double). Use a loop to read the 3 games.
3) After that it sells two games to customers. For each game, it reads the name of the game before selling it.
4) Finally, it prints the `totalSale`.

Answer

```
public class GameStore {

	String[] name;
	double[] price;
	int numOfGames;
	double totalSale;

	public GameStore(){
		this(100);
	}
	public GameStore(int inventorySize){
		name = new String[inventorySize];
		price = new double[inventorySize];
		numOfGames = 0;
	}
	public void add(String name, double price){
		if (findGame(name) == -1){
			if (numOfGames < this.name.length){
				this.name[numOfGames] = name;
				this.price[numOfGames] = price;
				numOfGames++;
			}
			else
				System.out.println("ERROR: NO ENOUGH SPACE IN INVENTORY");
		}
		else
			System.out.println("ERROR: GAME ALREADY IN STORE");
	}
	public int findGame(String name){
		for (int i = 0; i < numOfGames; i++){
			if (this.name[i].equals(name))
				return i;
		}
		return -1;
	}

	public void sell(String name){
		int i = findGame(name);
		if ( i != -1){
			totalSale += this.price[i];
			this.name[i] = this.name[numOfGames];
			this.price[i] = this.price[numOfGames];
			numOfGames--;
		}
		else
			System.out.println("ERROR: GAME IS SOLD OUT");
	}
	public double getTotalSale(){
		return totalSale;
	}
}
```

```
import java.util.Scanner;
public class TestGameStore {

	public static void main(String[] args) {
		Scanner s = new Scanner(System.in);
		GameStore gs = new GameStore(s.nextInt());
		String name;
		double price;

		for (int i = 0; i < 3; i++){
			System.out.print("Enter name of game: ");
			name = s.next();
			System.out.print("Enter price of game: ");
			price = s.nextDouble();
			gs.add(name, price);
		}

		System.out.print("Enter name of game: ");
		name = s.next();
		gs.sell(name);

		System.out.print("Enter name of game: ");
		name = s.next();
		gs.sell(name);

		System.out.println("Total sale price is: " + gs.getTotalSale());
	}

}
```

+ Bonus Question (5 Marks)

Write a Java method `sum` that takes an array of integers ***a*** and returns a **new array** of integers ***b*** of **same size** such that each element ***i*** in ***b*** is the result of addition (summation) of all elements ***0*** to ***i*** in ***a***.

Example: Given ***a*** = [1, 3, 2, 6, 10], method `sum` returns array ***b***=[1, 4, 6, 12, 22] since 1 = 1, 4 = 1+3, 6 = 1+3+2, 12 = 1+3+2+6, 22 = 1+3+2+6+10.

Answer:

```
import java.util.Arrays;

public class SumTest {

	public static void main(String[] args) {
		System.out.println(Arrays.toString(
					sum(new int[]{1, 3, 2, 6, 10})
				));

	}

	public static int[] sum(int[] a){
		int[] b = new int[a.length];
		for (int i = 0; i < b.length; i++){
			int total = 0;
			for (int j = 0; j <= i; j++){
				total += a[j];
			}
			b[i] = total;
		}
		return b;
	}

}
```

Result					
Question No.	**Relevant Student Outcome**	**SO is Covered by %**	**Full Mark**	**Student Mark**	**Assessor's Feedback**
1	a	25	11		
2	a	25	10		
3	a	10	4		
4	c	40	15		
Bonus Question	c	15	5		
Totals		100% + 15%	40 + 5		

I certify that the work contained within this assignment is all my own work and referenced where required. **Student Signature:** **Date:**			**Feedback Received:** **Student Signature:** **Date:**	