

**Question 1: (12 Marks)**

a- what is the output made by the following portion of code

```
int i =1, V=0, W=0;
while (i<=10 && V-W<=10) {
    W = V ;
    V = 2* V + i;
    i++;
    System.out.println("V =" +V);
}
```

**Answer :**

```
V =1
V =4
V =11
V =26
```

b- Suppose that we have an array **Marks** that has the length 5 and the following content

**Marks =**

10	20	15	17	9
----	----	----	----	---

- What is the output of the following code segment:

```
int s =0 ;
for(int i=0;i<Marks.length-1;i++){
    if (Marks[i]< Marks[i+1]){
        System.out.println("a["+i+"]="+
Marks[i]);
        s = s + Marks[i];
    }
    System.out.println("s = " + s);
}
```

**Answer :**

```
a[0]=10
a[2]=15
s = 25
```

- What is the output of the following code segment:

```
int i = 0, s = 0, c = 0;
while (2-c>0){
    if (Marks[i]>15) {
        c++;
        System.out.println("c = " +c);
    }
    i++;
    s += c;
}
System.out.println("result="+Marks[i]+" s="+s);
```

**Answer :**

```
c = 1
c = 2
result=9 s=4
```

**Question 2: Given the following class Bicycle (12 marks)**

<b>public</b> class Bicycle {
<b>private</b> int gear;
<b>private</b> int speed;
<b>public</b> Bicycle(int startSpeed, int startGear) {
gear = startGear;
speed = startSpeed;
}
<b>public</b> int getGear() {
return gear;
}
<b>public</b> void setGear(int newValue) {
gear = newValue;
}
<b>public</b> int getSpeed() {
return speed;
}
<b>public</b> void applyBrake(int decrement) {
speed -= decrement;
}
<b>public</b> void speedUp(int increment) {
speed += increment;
}
<b>public</b> boolean equal(Bicycle B) {
return (gear == B.gear && speed == B.speed);
}
}

**complete the needed statements in the Main class according to the following :**

```
public class Main {  
    public static void main(String[] args){  
        //1.     Declare and create two objects of type Bicycle:  
        //     - C1 has an initial gear of 4 and an initial speed of 20,  
        //     - C2 has an initial gear of 2 and an initial speed of 15.  
        Bicycle c1 = new Bicycle(20,4);  
        Bicycle c2 = new Bicycle(15,2);  
    }  
}
```

//2.      **Write the statement(s) to speedup object C2 with 5 units.**

```
c2.speedUp(5);
```

//3.      **Write the statements to copy C2 into C1.**

```
int c2Speed = c2.getSpeed();
int c2Gear= c2.getGear();
c1 = new Bicycle(c2Speed,c2Gear);
```

```
// or
```

```
// int gear = c2.getGear();
// int speed = c2.getSpeed();
// int oldSpeed = c1.getSpeed();
// c1. applyBrake(oldSpeed);
// c1.setGear(gear);
// c1.speedup(speed);
```

```
// or
```

```
// if (c2.getSpeed() > c1.getSpeed() )
//   c1.speedUp(c2.getSpeed() - c1.getSpeed() )
// else
//   c1. applyBrake( c1.getSpeed() - c2.getSpeed() )
```

//4.      **Write the statement(s) that displays 'YES' if C1 and C2 are equal.**

```
if ( c1.equal(c2) == true )
```

```
System.out.print("yes");
```

```
}  
}
```

**Question 3 : (16 Marks)****a- Complete the methods of the following class (11 Marks)**

```
public class MobileStore {

    private int[]    codes ; // stores the code of the Mobile
    private double[] prices ; // stores the price of the Mobile
    private int[]    quantities ; // stores the quantity of Mobile
    private int counter ; // counts the number of the inserted Mobiles

    public MobileStore (int size) {
        codes    = new int[size];
        prices    = new double[size];
        quantities = new int[size];
        counter = 0 ;
    }

    public double getMobilePrice(int index) {
        //This method returns the price of the Mobile located at index

        if ( index < 0 || index >= counter )
            return -1;
        else
            return prices[index];

    }

    public void insertMobile(int id, double pce, int qtity) {
        // this method adds, if possible, the given data
        //(id, pce and qtity) related to a new mobile

        if ( counter >= prices.length )
            System.out.print("NoSpace");
        else
        {
            codes[counter] = id;
            prices[counter] = pce;
            quantities[counter] = qtity;
            counter++;
        }
    }
}
```

```
}
```

```
public int cheapestMobile() {
```

```
    //returns the code of the available Mobile that has the minimum  
    // price. (available means that the quantity >0)
```

```
    int cheapestCode = -1;  
    double min = prices[0];  
    for ( int i = 0 ; i < counter ; i++){  
        if ( prices[i] <= min && quantities[i] > 0){  
            min = prices[i];  
            cheapestCode = codes[i] ;  
        }  
    }  
    return cheapestCode;
```

```
}
```

```
public boolean isMobileAvailable(int cd) {
```

```
    // if a Mobile with code cd exists and its quantity is greater than  
    // zero then return true, and otherwise return false.
```

```
    for ( int i = 0 ; i < counter ; i++){  
        if ( codes[i] == cd && quantities[i] > 0)  
            return true;  
    }  
    return false;
```

```
}

public void addQuantity(int cd, int newQuantity) {
    // newQuantity is added to the quantity of the Mobile
    // that has the code cd (newquantity MUST be a positive number)

    if (newQuantity > 0 ){
        for ( int i = 0 ; i < counter ; i++){
            if ( codes[i] == cd )
                quantities[i]+=newQuantity;
        }
    }
}
```

b- Give the UML representation of the class **MobileStore** (5 Marks)

MobileStore
<ul style="list-style-type: none"><li>- codes: int []</li><li>- prices: double[]</li><li>- quantities: int[]</li><li>- counter : int</li></ul>
<ul style="list-style-type: none"><li>+ MobileStore(size:int)</li><li>+ getMobilePrice (index:int):double</li><li>+ insertMobile(id:int, pce:double, qtity:int):void</li><li>+ cheapestMobile ():int</li><li>+ isMobileAvailable(cd:int):boolean</li><li>+ addQuantity(cd:int, newQuantity:int):void</li></ul>



# King Saud University

College of Computer and Information Sciences  
Computer Science Department

<b>Course Code:</b>	CSC 111
<b>Course Title:</b>	Computer Programming I
<b>Semester:</b>	Fall 2011-2012
<b>Exam:</b>	<b>Final Exam</b>
<b>Duration:</b>	3 hours

Student Name:

Student ID:

Student Section No.

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;	<b>Part I</b>	<b>25</b>
	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;	<b>Part II, III</b>	<b>75</b>
	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.		



Part I. (10 Marks)

I.1 What is the printout of the loop?

```
int i = 0;
while (i < 10) {
    if ((i + 1) % 2 == 0)
        System.out.println(i);
    i++;
}
```

Answer I.1: (4 Marks)

1	0.00
3	0.00
5	0.00
7	0.00
9	0.00

I.2 Suppose the input is 2 3 5 4 0. What is the output of the following segment code?

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

int number , value =0;
number = input.nextInt();

while (number != 0) {
    if (number > value)
        value = number;
    number = input.nextInt();
}
System.out.println("Value is " + value);
System.out.println("number " + number);
}
```

Answer I.2: (3 Marks)

Value is 5	1.50
number is 0	1.50

**I.3 Convert the following 'if statement' using a 'switch' statement**

```
// Find interest rate based on year
if (numOfYears == 7)
    annualInterestRate = 7.25;
else if (numOfYears == 15)
    annualInterestRate = 8.50;
else if (numOfYears == 30)
    annualInterestRate = 9.0;
else {
    System.out.println("Wrong number of years");
}
```

**Answer I.3: (3 Marks)**

```
switch (numOfYear) { ..... 0.00
    case 7: annualInterestRate = 7.25; ..... 0.00
        break; ..... 0.00
    case 15: annualInterestRate = 8.50; ..... 0.00
        break; ..... 0.00
    case 30: annualInterestRate = 9.0; ..... 0.00
        break; ..... 0.00
    default: System.out.println("Wrong number of years"); ..... 0.00
}
```

Part II. Complete the methods of the following class  
(20 Marks: 2marks for the first method and 3 marks for each of the other methods)

```
public class Game100 {  
    private int[] itemsCodes ;    // stores the code of the items  
    private double[] itemsPrices ; // stores the price of the items  
    private int[] itemsQuantities ; // stores the quantity of the items  
    private int counter ; // counts the number of the inserted items  
  
    public Game100 (int size) {    // this constructor is given  
        itemsCodes = new int[size];  
        itemsPrices = new double[size];  
        itemsQuantities = new int[size];  
        Counter = 0 ;  
    }  
  
    public int getCounter() { // return the value of counter  
        return counter; .....2.00  
    }  
  
    public void insertItem(int code, double price, int quantity) {  
        // if the counter does not exceed the size of the array  
        // inserts the data (code, price and quantity) of this new item  
        // we suppose that the code of the item was not inserted before  
  
        if (counter < itemsCodes.length) { .....0.75  
            itemsCodes[counter] = code; .....0.50  
            itemsPrices[counter] = price; .....0.50  
            itemsQuantities[counter] = quantity; .....0.50  
            counter++; .....0.75  
        }  
    }  
}
```



```

public int findCheapestItem() {
    //returns the Code of the item that has the minimum price.

    int min_price = Integer.MIN_VALUE; .....0.00
    int min_code = -1; .....0.00

    for (int i=0; i<counter; i++) .....0.00
        if (itemsPrices[i] < min_price) { .....0.00
            min_price = itemsPrices[i]; .....0.00
            min_code = itemsCodes[i]; .....0.00
        }

    return min_code; .....0.00

}

public boolean isItemAvailable(int codeItem) {
    // if the item with codeItem exists and its Quantity
    // is greater than zero it returns true otherwise it returns false.

    for (int i=0; i<counter; i++) .....0.00
        if (itemsCodes[i]==codeItem && itemsQuantities[i]>0) .....0.00
            return true; .....0.00
    return false; .....0.00

}

```

```

public double getItemPrice(int index) {
    //returns the price of the item located at index

    if (index >=0 && index < counter) .....0.00
    return itemsPrices[index]; .....0.00
}

public void sellItem(int wantedCodeItem, int wantedQuantity) {
    // if the item that has a code = wantedCodeItem exists
    // and it has a quantity greater or equal to wantedQuantity,
    // it modifies its itemsQuantities value.

    for (int i=0; i<counter; i++) .....0.00
        if (itemsCodes[i] == wantedCodeItem &&
            itemsQuantities[i] >= wantedQuantity) { .....0.00
            itemsQuantities[i] -= wantedQuantity; .....0.00
            return; // depends on the requirement .....0.00
        }

}

public void addQuantity(int codeItem, int newQuantity) {
    // newQuantity is added to the quantity of the item
    // that has the code= codeItem

    for (int i=0; i<counter; i++) .....0.00
        if (itemsCodes[i] == wantedCodeItem) { .....0.00
            itemsQuantities[i] += wantedQuantity; .....0.00
            return; // depends on the requirement .....0.00
        }

}

```

Part III. Complete the following Java program that uses Game100 to do the following:

- a- Create an object of the class Game100 that can process 200 items
- b- Insert the item that has the following data:  
Code = 12345, price = 24.95, quantity = 1200
- c- Insert one item where its data (code, price and quantity) is entered by the user
- d- Add 20 to the quantity of the cheapest item
- e- Display the prices of all the inserted items

Answer part III: (10 Marks: 2 marks for each question)

```
import java.util.Scanner;

public class TestGame100 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        Game100 obj = new Game100(200); .....2.00

        Obj.insertItem(12345, 24.95, 1200); .....2.00

        Obj.insertItem(input.nextInt(),
                       input.nectDouble(),
                       input.nectInt()); .....2.00

        int cheapestCode = obj.findCheapestItem(); .....1.00
        obj.addQuantity(cheapestCode, 20); .....1.00

        for (int i=0; i< obj.getCounter(); i++) .....1.00
            System.out.println(obj.getItemPrice(i)); .....1.00

        // or, simply:
        // System.out.println(obj.getItemPrice(0));
        // System.out.println(obj.getItemPrice(1));

    }
```

Result					
Question No.	Relevant Student Outcome	SO is Covered by %	Full Mark	Student Mark	Assessor's Feedback
I	a	10%	4		
I	a	7.5%	3		
I	a	7.5%	3		
II	c	50%	20		
III	c	25%	10		
Totals		100%	40		
<p><b>I certify that the work contained within this assignment is all my own work and referenced where required.</b></p> <p><b>Student Signature:</b> _____ <b>Date:</b> _____</p>					<p><b>Feedback Received:</b></p> <p><b>Student Signature:</b> _____</p> <p><b>Date:</b> _____</p>







# King Saud University

College of Computer and Information Sciences

Computer Science Department

**Course Code:**

CSC 111

**Course Title:**

Introduction to Programming

**Semester:**

Fall 2012

**Exercises Cover  
Sheet:**

**Final  
Exam**

**Duration : 3 hours**

Instructor:

Student Name:

Student ID:

Student Section No.

Tick the Relevant	Computer Science B.Sc. Program ABET Student Outcomes	Question No. Relevant Is Hyperlinked	Coverin g %
√	a) Apply knowledge of computing and mathematics appropriate to the discipline;	1,2	35
	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;	3,4	65
	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: (6 Marks)**

Give the UML class representation of the following Java class:

```
public class Map {
    private int [] StreetId;    private String [] StreetName;
    private String [] shape;    private int counter;
    static private int id ;

    //-----
    public Map(int size) {
        StreetId = new int[size];    StreetName = new String [size];
        counter = 0;    id = 0 ;        shape = new String[size];
    }

    //-----
    private int searchStreet(String name) {
        for (int i=0; i<counter; i++) if (StreetName [i].equals(name)) return i;
        return -1;
    }

    //-----
    public boolean addStreet(String name) {
        int i = searchStreet (name);
        if (i == -1 && counter< StreetId.length) {
            StreetName[counter] = name;    StreetId[counter] = id; shape[counter]=null ;
            counter++;    id++ ;    return true;
        }
        return false;
    }

    //-----
    public boolean removeStreet(String name) {
        int i = searchStreet (name);
        if (i != -1) {
            StreetName [i] = StreetName [counter-1];
            StreetId [i] = StreetId [counter-1];
            shape[i] = shape[counter-1] ; counter-- ;
            return true;
        }
        return false;
    }

    //-----
    public void displayStreetAtIndex(int index) {
        if (index>=0 && index<counter)
            System.out.println(StreetName[index]+ "--"+StreetId[index]+ "--"+
            shape[index]);
    }

    //-----
    public String getStreetName(int index) {
        if (index>=0 && index<counter) return StreetName[index];
        return "out of range";
    }
}
```



**Answer of Question 1:****Map**

- StreetId: int []
- StreetName: String []
- shape: String []
- counter: int
- id: int
- + Map(in size:int): void
- searchStreet(in name:String): int
- + addStreet(in name:String): Boolean
- + removeStreet(in name:String): boolean
- + displayStreetAtIndex(in index: int): void
- + getStreetName(in index: int): String

**Question 2 : (12 marks)**

Suppose that we have an array Marks that has the length 5 and the following content

```

          10    20    15    17
              9
Marks =

```

a. What is the output of the following code segment:

```

int s =0 ;
for (int i=0; i<Marks.length-1(=4); i++)
    if (Marks[i]< Marks[i+1]){
        System.out.println("a["+i+"]="+ Marks[i]);
        s = s + Marks[i];
    }
System.out.println("s = " + s);

```

```

Loop1 0<4: i=0, i+1=1, if(10<20)=true, s= 0+10=10
Loop2 1<4: i=1, i+1=2, if(20<15)=false, s=10
Loop3 2<4: i=2, i+1=3, if(15<17)=true s=10+15=25
Loop4 3<4: i=3, i+1=4, if(17<9)=false s=25
4<4 → Stop the loop

```

**Answer :**

a[0]=10

a[2]=15

s = 25

b. What is the output of the following code segment:

```

int i = 0, s = 0, c = 0;
while (2-c>0){
    if (Marks[i]>15) c++;
    i++;
    s += c;
}
System.out.println("result="+Marks[i]+" s="+s);

```

```

Loop1 2-0>0: i=0, 10>15=false, i=1, s=0+0=0; c=0;
Loop2 2-0>0: i=1, 20>15=true, c=1, i=2, s=0+1=1;
Loop3 2-1>0: i=2, 15>15=false, i=3, s=1+1=2;
Loop4 2-1>0: i=3, 17>15=true, c=2; i=4, s=2+2=4;
Loop4 2-2>0: false → stop the loop
i=4, s =4
System.out.println("result="+Marks[4]+" s="+4);

```

**Answer :**

result=9 s=4

c. What is the output of the following code segment:

```
int i = Marks.length (=5), x = 0;
while (i>1 && x<=0) {
    Marks [i-2] (=3,2,1,0) = Marks [i-1]+ x;
    x = x * 2 + i;
    i--;
    x = (x-i-1)/2;
}
System.out.println("mark0 =" + Marks[0]);
System.out.println("mark2 =" + Marks[2]);
```

```
i=5, x=0
Loop1 5>1 && x<=0: Marks[3]=Marks[4]+0=9+0=9, x=0*2+5=5, i=4,
x=5-4-
1/2=0/2=0; Marks={10,20,15,9,9}
Loop2 4>1 && x<=0: Marks[2]=Marks[3]+0=9+0=9, x=0*2+4=4, i=3,
x=4-3-
1/2=0/2=0; Marks={10,20,9,9,9}
Loop3 3>1 && x<=0: Marks[1]=Marks[2]+0=9+0=9, x=0*2+3=3, i=2,
x=3-2-
1/2=0/2=0; Marks={10,9,9,9,9}
Loop4 2>1 && x<=0: Marks[0]=Marks[1]+0=9+0=9, x=0*2+2=2, i=1,
x=2-1-
1/2=0/2=0; Marks={9,9,9,9,9}
Loop5 1>1 && x<=0: false → stop the loop
Marks[0]=20, Marks[2]=17
```

**Answer:**

mark0 =9

mark2 =9

d. What is the number of iterations made by the portion of code (c)

4 iterations.//not sure of the answer

**Question 3: (12 Marks)**

Suppose that we have a ToyShop class (see the UML class diagram):

<b>ToyShop</b>	
- nameOfToy : String[] - price : integer[] - quantity : integer[] - nbToys : integer	
+ <b>ToyShop</b> (capacity : integer ) + <b>addToy</b> (toyName:String, p:integer, quant:integer): boolean + <b>searchToy</b> (toyName:String):integer + <b>deleteToy</b> (toyName : String) : boolean + <b>displayAll</b> () : void + <b>displayGivenToy</b> (toyName : String) : void + <b>displayNameOfToyAtIndex</b> (index : integer) : void + <b>mostExpensiveToy</b> () : integer + <b>displayRangeOfToys</b> (price1 : integer, price2 : integer) : void + <b>getNbToys</b> ( ) : integer	

**Attributes :**

- **nameOfToy** : array that stores name of toys in the store (e.g. doll, robot, car etc)
- **price** : array that stores prices of toys in the store.
- **quantity** : array that stores quantity of each toy in the store.
- **nbToys** : Total number of toys stored in toy shop.

**Methods :**

- **ToyShop**(capacity: int ) : constructor method which creates arrays with the given capacity(the maximum number of toys that the Toy Shop can store).
- **addToy**(toyname:String,p:int, quant:int): adds a new toy in the toy store with the name *toyName*, price *p* and quantity *quant*. This method will return true if toy is added successfully otherwise false.
- **searchToy**(toyName : String): searches a toy whose name is *toyName*, returns its index (cell number) on successful search otherwise -1.
- **deleteToy**(toyName : String): deletes a toy whose name is *toyName*, returns true on successful deletion otherwise false.
- **displayAll**( ) : Shows data of toys in the store.
- **displayGivenToy**(toyName : String) : Shows data of the toy whose name is *toyName*.
- **displayNameOfToyAtIndex**(index : integer) : Shows name of the toy whose is at the cell *index*.
- **mostExpensiveToy**( ) : returns the price of the most expensive toy.



- **displayRangeOfToys**(price1 : integer, price2 : integer) : display all toys information having a price comprises between price1 and price2.
- **getNbToys**( ) : returns total number of toys in store.

**Question** : implement the class **ToyShop**

**Answer of Question 4 (Part A):**

```
public class ToyShop{
private String[]nameOfToy;
private int[] price;
private int[]quantity;
private int nbToys;
public ToyShop(int capacity){
    nbToys=0;
    nameOfToy= new String[capacity];
    price= new int [capacity];
    quantity= new int [capacity];
}
public boolean addToy(String toyName, int p, int quant){
    if (nbToys< nameOfToy.length) {
        nameOfToy[nbToys] = toyName;
        price[nbToys] = p;
        quantity[nbToys]=quant ;
        nbToys++;
        return true;
    }
    return false;
}
public int searchToy(String toyName){
    for ( int i=0; i< nameOfToy.length;i++)
        if (nameOfToy[i].equals(toyName)==true)
            return i;
    return -1;
}
public boolean deleteToy(String toyName){
    for ( int i=0; i<nbToys;i++)
        if (nameOfToy[i].equals(toyName)==true){
            nameOfToy [i] = nameOfToy [nbToys -1];
            price [i] = price [nbToys -1];
            quantity [i] = quantity [nbToys -1] ; nbToys -- ;
            return true; }
    return false;
}
public void displayAll(){
    for ( int i=0; i< nbToys;i++)
        System.out.println(nameOfToy [i]+ "--"+price[i]+ "--"+ quantity[i]);
}
public void displayGivenToy(String toyName){
    for ( int i=0; i< nbToys;i++)
        if (nameOfToy[i].equals(toyName)==true){
            System.out.println(nameOfToy [i]+ "--"+price[i]+ "--"+ quantity[i]);
        }
}
public void displayNameOfToyAtIndex(int index){
    System.out.println(nameOfToy [index]);
}
public int mostExpensiveToy(){
    int expensive = price[0];
    for ( int i=1; i< nbToys;i++)
        if (price[i]> expensive){
            expensive= price [i];}
    return expensive;
}
public void displayRangeOfToys(int price1, int price2){
    for ( int i=1; i< nbToys;i++)
        if (price[i]>= price1 && price[i]<= price2 ){
            Svsystem.out.println(nameOfToy [i]+ "--"+price[i]+ "--"+ quantity[i]); }
```



Using the class ToyShop defined in question 3, Write a Java program **ToyShopTest** that performs the following processing:

- Creates the ToyShop object with a capacity **entered by the user**.
- Adds the following Toys in the Shop

Toy Name	Price	Quantity
Doll	12	20
Robot	20	30
Balloon	30	30

- Inserts 3 toys entered by the user
- Deletes the toy "Balloon" from the toyShop.
- Displays information of the toy "Robot".
- Displays **ONLY** the name of all toys.

#### Answer of Question4 (Part B):

```
import java.util.*;
public class ToyShopTest{
    public static void main(String[]arg){
        Scanner input = new Scanner(System.in);
        System.out.println("Enter the size of the array: ");
        int size= input.nextInt();
        ToyShop toyshop = new ToyShop(size);
        boolean b1=toyshop.addToy ("Doll",12,20);
        boolean b2=toyshop.addToy ("Robot",20,30);
        boolean b3=toyshop.addToy ("Balloon",30,30);
        String name; int price; int quant; boolean b;
        for(int i=1;i<=3;i++){
            System.out.println("Enter the name of the toy: ");
            name= input.next();
            System.out.println("Enter the price of the toy: ");
            price= input.nextInt();
            System.out.println("Enter the quantity of the toy: ");
            quant= input.nextInt();
            b= toyshop.addToy (name,price,quant);
        }
        boolean bool= toyshop.deleteToy("Balloon");
        toyshop.displayGivenToy("Robot");
        int capacity= toyshop.getNbToys();
        for (int i=0; i< capacity; i++)
            toyshop.displayNameOfToyAtIndex(i);
    }
}
----jGRASP exec: java ToyShopTest
```

Enter the size of the array:

10

Enter the name of the toy:

t1

Enter the price of the toy:

3  
Enter the quantity of the toy:  
5  
Enter the name of the toy:  
t2  
Enter the price of the toy:  
13  
Enter the quantity of the toy:  
7  
Enter the name of the toy:  
t3  
Enter the price of the toy:  
22  
Enter the quantity of the toy:  
15  
Robot--20--30  
Doll  
Robot  
t3  
t1  
t2  
  
----jGRASP: operation complete.

Result					
Question No.	Relevant Student Outcome	SO is Covered by %	Full Mark	Student Mark	Assessor's Feedback
1	a	15	6		
2	a	30	12		
3	c	30	12		
4	c	25	10		
Totals		100%	40		

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



# King Saud University

College of Computer and Information Sciences

Computer Science Department

**Duration :** 3 hours

**Course Code:** CSC 111

**Course Title:** Introduction to Programming

**Semester:** Fall 2014

**Exercises Cover Sheet:** **Final Exam**

Student Name:

Student ID:

Student Section No.

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;	2	30
	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;	1,3,4	70
	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: (8 Marks)**

a- Draw the UML class diagram of the following Java code:

```
public class NamesList {
    private String name[];
    private int code[];
    private int number;

    public NamesList(int z){
        name = new String[z];
        code = new int[z];
        number = 0;
    }
    public boolean AddNewRec(int ident, String mn){
        if (number<code.length){
            code[number]=ident;
            name[number] = new String(mn);
            number++;
            return true;
        }
        else
            return false;
    }
    public void display(){
        int i;
        for (i=0;i<number;i++){
            System.out.println(code[i]+" --- "+name[i]);
        }
    }
    public String display(int location){
        if (location>=0 && location<number)
            return name[location];
        else
            return "not present";
    }
    public void deleteByLastIndex(int location){
        if (location>=0 && location<number){
            code[location]= code[number-1];
            name[location]=name[number-1];
            number--;
        }
    }
    public int deleteByShiftingData(int location){
        if (location>=0 && location<number){
            for (int i= location; i<number-1; i++){
                code[i]= code[i+1];
                name[i]=name[i+1];
            }
            number--;
        }
        return number;
    }
}
```

Answer:

<b>NamesList</b>	.....	0.50
- Name : string[]	.....	0.50
- Code : integer[]	.....	0.50
- Number : int	.....	0.50
+ NamesList(integer)	.....	1.00
+ AddNewRec(integer, string) : bool	.....	1.00
+ display()	.....	1.00
+ display(integer): string	.....	1.00
+ deleteByLastIndex(integer)	.....	1.00
+ deleteByShiftingData(int)	.....	1.00

**Question 2: (12 Marks)**

I) Answer the following with True/False:

Question	Answer (True/False)
a. The output of the following code segment is "26": <code>int someInt=26; System.out.println("someInt");</code>	False.. 1
b. To declare a boolean variable b in Java we write: <code>boolean b;</code>	True... 1
c. In java, a variable x is divisible by 5 if the expression <code>(x%5==0)</code> is evaluated true.	True... 1
d. Using a <i>switch</i> -statement only (without using if), we can know whether an integer x is equal to 0 or not.	True... 1
e. Compiling a program successfully guarantees that it executes correctly.	False.. 1
f. The following two lines of code do the same thing: <code>for(i=1;i&lt;5;i++) z = z+i;</code> AND <code>i=1; do { z += i; i = i + 1; } while (i&lt;5);</code>	True... 1
g. The following statement is valid in Java <code>int x=1; double y=2.0; x = x+y;</code>	False.. 1
h. The expression <code>(x = 0)</code> is used in an <i>if</i> -statement to know whether x is equal to 0.	False.. 1

II) Multiple choice: Circle only one choice in each question:

a. After executing the following code segment:

`int result=1; for(i=1;i<=4;i++) result = result*i;`

the variable result will contain the value:

- a) 20
- b) 24 ..... 1.00
- c) 30
- d) 10

b. Given two variables x and y of type int, where we assume that x=3, and y=4, we calculate z as follows:

`int z=0; for(i=0;i<x;i++) z += y;`

what will the value of z be?

- a) 0
- b) 3
- c) 8
- d) 12 ..... 1.00
- e) Null

c. What will the value of v be after executing the following code segment?

Suppose that the value stored in 'a' are

5	8	3	0
---	---	---	---

```
v=a[0];
```

```
for(i=1;i<a.length;i++) {if (a[i]>v) v=a[i]};
```

- a) 0
- b) 8 ..... 1.00
- c) 4
- d) 5
- e) Null

d. Given a Boolean variable w, which of the following is a valid assignment statement?

- a) w = false;
- b) w = w && w;
- c) w = 5 > n;
- d) a and c above
- e) a, b, and c above ..... 1.00

e. What is the output of the following code segment:

```
System.out.print(10/3);
```

- a) 3 ..... 1.00
- b) 3.3
- c) 3.33333333
- d) none of the above

**Question 3 : (10 Marks)**

Implement in Java the following CourseSection class diagram:

CourseSection
- courseName : String - nameOfStudent : String[] - studentID : integer[] - GPA : double[] - nbStudents : integer
+ CourseSection(capacity : integer, cName: String) + addNewStudent(stName:String, stID:integer, stGPA:double):boolean + getStudentIndex(stId : integer) : integer + getnbStudents() : integer + deleteStudent(stId : integer) : boolean + displayStudentAtIndex(k : integer) : void + displayPassOrFailStudents (indicator : String) : void

**Attributes :**

- nameOfStudent : Array that stores names of students registered in the section.
- studentID : Array that stores IDs of students registered in the section.
- GPA : Array that stores gpa of each student registered in the section.
- nbStudents : Number of students registered in the section currently.
- courseName : Name of the course (e.g. Programming 1)

**Methods :**

- a) CourseSection(capacity : integer, cName: String): A constructor method which creates arrays with the given capacity (the maximum number of students that a section can have) and assigns cName to courseName and initializes the attributes nbStudents to zero.
- b) addNewStudent(stName:String, stID:integer, stGPA:double) : Adds a new student in the section with the studentName as stName, studentID as stId and GPA as stGPA. This method returns true if the student is added successfully otherwise it returns false.
- c) getStudentIndex(stId : integer): returns index of the student whose ID is stId otherwise -1.
- d) getnbStudents() : returns total number of student currently registered in the section.
- e) deleteStudent(stId : integer): deletes a student whose ID is stId. This method returns true on successful deletion otherwise it returns false.
- f) displayStudentAtIndex(k : integer): Shows data of the student whose is stored at position k.
- g) displayPassOrFailStudents(indicator : String) : void: Shows information of all the students in the section whose indicator is according to the following table:

indicator	GPA
Pass	$\geq 2.5$
Fail	$< 2.5$

If the indicator is equal to Pass then the method would display ONLY the students who passed. If the indicator is equal to Fail then the method would display ONLY the students who failed.



**Answer:**

```

class CourseSection {
    private String courseName; ..... 1.00
    private int nbStudents; .....0.25
    private String[] nameOfStudent; .....0.25
    private int[] studentID; .....0.25
    private double[] GPA; .....0.25

    public CourseSection(int capacity, String cName) { .....0.25 1.50
        courseName = cName; .....0.25
        nameOfStudent = new String[capacity]; .....0.25
        studentID = new int[capacity]; .....0.25
        GPA = new double[capacity]; .....0.25
        nbStudents = 0; .....0.25
    }

    public boolean addNewStudent(String stName, int stID, double stGPA) { .0.25 2.00
        if (nbStudents == nameOfStudent.length) return false; .....0.25
        if (getStudentIndex(stID) != -1) return false; .....0.25
        nameOfStudent[nbStudents] = stName; .....0.25
        studentID[nbStudents] = stID; .....0.25
        GPA[nbStudents] = stGPA; .....0.25
        nbStudents++; .....0.25
        return true; .....0.25
    }

    public int getStudentIndex(int stID) { .....0.25 1.00
        for (int i=0; i < nbStudents; i++) .....0.25
            if (studentID[i] == stID) return i; .....0.25
        return -1; .....0.25
    }

    public int getNbStudents() { .....0.25 0.50
        return nbStudents; .....0.25
    }

    public boolean deleteStudent(int stID) { .....0.25 2.00
        int indx = getStudentIndex(stID); .....0.25
        if (indx == -1) return false; .....0.25
        nbStudents--; .....0.25
        nameOfStudent[indx] = nameOfStudent[nbStudents]; .....0.25
        studentID[indx] = studentID[nbStudents]; .....0.25
        GPA[indx] = GPA[nbStudents]; .....0.25
        return true; .....0.25
    }

    public void displayStudentAtIndex(int k) { .....0.25 1.00
        if (k >= 0 && k < nbStudents) { .....0.25
            System.out.println( "Student's name:" + nameOfStudent[k] + "\n" +
                               "Student's ID:" + studentID[k] + "\n" +
                               "Student's GPA:" + GPA[k] ); .....0.50
        }
    }

    public void displayPassOrFailStudents(String indicator) { .....0.25 1.00
        for (int i=0; i<nbStudents; i++) .....0.25
            if ( (indicator.equals("Pass") && GPA[i] >= 2.5) ||
                (indicator.equals("Fail") && GPA[i] < 2.5) ) .....0.25
                displayStudentAtIndex(i); .....0.25
    }
}

```



**Question 4 : (10 Marks)**

Using class `CourseSection`, write a Java class `CourseSection`, which contains a main program that performs following operations:

- Create `CourseSection` object with capacity of 45 student and course name "Programming 1".
- Add following students in the section.

Student Name	ID	GPA
Ali	42910011	2.6
Fahad	43121476	2.1
Majid	43395138	3.5

- (we suppose that we inserted several students) Delete student whose ID is 43395138 from the section.
- Display information about the student whose ID is 43122345.
- Display all information about all students.
- Display all information about all students whose grade is "Pass".

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

        //create CourseSection object here. (part a)
        CourseSection cs = new CourseSection(45, "Programming 1"); ..... 1.25

        //add 3 students as given in the table above (part b).
        cs.addNewStudent("Ali", 42910011, 2.6); ..... 0.75
        cs.addNewStudent("Fahad", 43121476, 2.1); ..... 0.75
        cs.addNewStudent("Majid", 43395138, 3.5); ..... 0.75

        //assume now there are many registered students in the section.
        //delete student whose ID is 43395138 from the section. (part c)
        cs.deleteStudent(43395138); ..... 1.00

        // display information about the student whose ID is 43122345. (part d)
        int i = cs.getStudentIndex(43122345); ..... 1.00
        cs.displayStudentAtIndex(i); ..... 1.00

        //display all information about all students. (part e)
        int nb = cs.getNbStudents(); ..... 0.75
        for(i = 0; i<nb; i++) ..... 0.75
            cs.displayStudentAtIndex(i); ..... 0.75

        //display all info about all students whose grade is "Pass". (part f)
        cs.displayPassOrFailStudents("Pass"); ..... 1.25

    }
}
```



Result					
Question No.	Relevant Student Outcome	SO is Covered by %	Full Mark	Student Mark	Assessor's Feedback
1	a	20	8		
2	c	30	12		
3	c	25	10		
4	c	25	10		
Totals		100%	40		
I certify that the work contained within this assignment is all my own work and referenced where required.  <b>Student Signature:</b> _____ <b>Date:</b> _____					<b>Feedback Received:</b>  <b>Student Signature:</b> _____ <b>Date:</b> _____