King Saud University

College of Computer and Information Sciences
Computer Science Department

Course Code:	CSC 111
Course Title:	Computer Programming I
Semester:	Fall 2011-2012
Exam:	**Final Exam**
Duration:	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;	Part I	25
	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;	Part II, III	75
	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		

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:**