Question 1: Put your answers of the question 1 **(True or False)** in the following table. **(10 Marks)**

Question	1	2	3	4	5	6	7	8	9	10
Answer (T/F)										

1.	The following code segment will lead to a runtime error
	`String a = "aaa"; String b = "bbb"; a = b; b = null;`
	`System.out.println(a.charAt(0));`

2.	The following two methods can be implemented in a Java class
	`void method1(int i){ }`
	`void method1(double j){ }`

3.	The following statement is valid in Java: `String 2ndPlayer = "Mohammad";`
4.	The following code segment will print 7 `String a = "FinalExam"; System.out.println(a.lastIndexOf('a'));`
5.	The following two statements do the same thing: • `do{ i++; }while(false);` • `i++;`

6.	The class defined below is valid in Java:
	`public class A {`
	`A(int x){}`
	`A(double x){} }`

7.	The following code segment will lead to an infinite loop `for(int i=3 ; i != 9; i++) i*=2;`
8.	The value of `x` after executing the following code segment is 5 `int x = 3, y = 5; x = y = x;`
9.	The output of the following statement is csc1112019final `System.out.print("csc111" + 2000 + 19 + "final");`
10.	The following code segment is invalid in Java: `final int x = 5; x++;`

Question 2: Put your answers (**multiple choice questions**) in the following table. **(10 Marks)**

Question	1	2	3	4	5	6	7	8	9	10
Answer										

1) What is the output of the following code segment (if any)?

```
int [] a = {3, 2, 5, 9};
int i,j;
for(i = 0; i <= 2; i++) {
        if(a[i] < a[i+1]) {
                j = a[i];
                a[i]=a[i+1];
                a[i+1]= j;
        }
}
for(i = 0; i < 4; i++)
        System.out.print(a[i]);
```

[A] Compilation error [B] 3529 [C] 2359 [D] 3592

2) What is the output of the following program (if any)?

```
class ClassB{
  public ClassB() { System.out.print("C"); }
  private void method1() { System.out.print("A"); }
}
class ClassA{
  public void method2() {
        ClassB c2 = new ClassB();
        System.out.print("B");
        c2.method1();
  }
  public static void main(String[] args) {
        ClassA c1 = new ClassA();
        System.out.print("D");
        c1.method2();
  }
}
```

[A] Compilation error [B] DBCA [C] DBAC [D] DCBA

3) What is the output of the following program (if any)?

```
class Student{
   public String name;
   public Student(String n) { name=n;}
}
public class Section {
   public static void changeName(Student s) {
         s.name = "Sultan";
         s = new Student("Fahad");
         s.name = "Saad";
         System.out.print(s.name);
   }
   public static void main(String[] args) {
         Student student = new Student("Saud");
         changeName(student);
         System.out.print(student.name);
   }
}
```

[A] Compilation error [B] SaadSaad [C] SultanSaud [D] SaadSultan

4) What is the output of the following code segment (if any)?

```
if(true)
{
      int i = 1;
      System.out.print(i);
}
else
      System.out.print(++i);
```

[A] 1 [B] 2 [C] Compilation error [D] Runtime error

5) Which of the following statements is valid to create an array for 5 objects of Student class?

[A] Student [] a = new Student(5);

[B] Student [] a = new Student[5];

[C] Student a = new Student[5];

[D] Student a = new [5] Student();

6) **What is the output of the following program (if any)?**

```
public class Final {
        public static int a = 1;
        private int b = 2;
        public Final(){
                a++; b++;
        }
        public static void printBoth() {
                System.out.print(a + " " + b + " ");
        }

        public static void main(String[] args) {
                Final f1 = new Final(), f2 = new Final();
                f1.printBoth();
                f2.printBoth();
        }
}
```

[A] Compilation error [B] 3 3 4 3 [C] 2 3 1 2 [D] 2 3 3 3

7) **What is the output of the following code segment (if any)?**

```
class ClassA{
        public static double pow(int i) {
                return i * i;
        }
}
public class Final {
        public static void main(String[] args) {
                System.out.println(ClassA.pow(5));
}}
```

[A] Runtime error [B] 25.0 [C] 25 [D]Compilation error

8) **What is the output of the following program (if any)?**

```
public class Final {
        public double x = 1;
        Final(){ x++; }
        Final(double y){ this(2); x = y;}
        Final(int n){ x--; }
        public static void main(String[] args) {
                Final f1 = new Final(3.5);
                System.out.print(f1.x);
}}
```

[A] 2 [B] 0 [C] 3.5 [D] 2.5

9) **What is the output of the following code segment (if any)?**

```
boolean water = true;
while (water) {
    System.out.print(water);
    while (water) {
        water = false;
        System.out.print(water);
    }
}
System.out.print(water);
```

[A] infinite loop [B] truefalsetrue [C] truetruetrue [D] truefalsefalse

10) What is the output of the following code segment (if any)?

```
int i=0;
char a = 'A';
while (i != 2)
      switch(a) {
            case 'A': i++; a ='C';
            case 'B': i--; a = 'A'; break;
            case 'C': i+=2; break;
            default: i=2;
      }
System.out.print(a);
```

[A] Infinite loop [B] A [C] B [D] C

Question 3. (10 Marks)

3.1) **(2 marks)** Each code segment below contains one or more **compilation errors**. First, place an X in the column to the right of the line with the mistake. Then, rewrite the lines with mistakes in the box that follows:

A)

1	`public class Account {`	
2	`  private double balance;`	
3	`  private String name;`	
4	`  public Account(double x, String n) { name = n; balance = x;}`	
5	`  public int getBalance() { return balance; }`	
6	`  public static void main(String arg[]){`	
7	`  Account n = new Account("Ahmed", 1000.0);`	
8	`}}`	

Fixed statements (two problems):

B)

1	`int data [] = {1,2,3,4};`	
2	`for(int i = 2; i < data.length() ; i++)`	
3	`  System.out.println(data[i]);`	

Fixed statement (one problem):

C)

```
1  public class XOR {
2      private boolean xor(boolean a, boolean b) { return a != b; }
3      public static void main(String[] args) {
4          System.out.print(xor(true, false));
5  }}
```

Fixed statement (one problem):

3.2) **(3 marks)** Each one of the following methods is *missing some code segments*. Read and understand the description of each method. Then, *complete the implementation* by filling in the blanks.

A) The **toArrayOfChar** method takes a string n (assume n isn't null) and returns array of characters that represents the content of the received string. Example: when **toArrayOfChar("CS111")** is called, it will return the following array:

'C'	'S'	'1'	'1'	'1'

```
public static ..................... toArrayOfChar(String n) {

    ............................ a = new ..............................................;

    for(int i = 0; i < ............................ ; i++)

        ........................ = ......................................;

    return a;
}
```

B) The **isReversed** method takes two arrays of integers (a and b) and returns true if b is the reversed version of a. Otherwise, it returns false. Example: when **isReversed({1,2,3}, {3,2,1})** is called, it will return true. However, when **isReversed({1,2,3}, {2,1,3})** is called, it will return false.

```
public static ............................ isReversed(int []a, int []b) {
    if(a.length != b.length)
        return false;

    for(int i = 0, j =......................; i < a.length ; i++, j--)

        if(........................ != ........................)

            return ..................................;

    return ............................;
}
```

3.3) **(2 marks)** Each code segment below produces an output. Write the output of each code segment in the provided answer boxes.

A)

```
public class Final {
  double x=0, y=0;
  public void m1(double a, int b) { x++; y++; }
  public void m2(int a, int b) { x--; y--; }
  public void m1(double a, double b) { x+=2; y+=2; }
  public static void main(String[] args) {
        Final f1 = new Final();
        int x = 3, y= 2;
        double z = 3;
        f1.m1(x,y);
        System.out.println(f1.x + " , " + f1.y);
        f1.m1(y, z);
        System.out.println(f1.x + " , " + f1.y);
  }
}
```

Answer (Output):

B)

```
public class FinalExam {
  public static void method(double array[]) {
        for(int i= 0; i< array.length; i++)
              if(i % 2 == 0)
                    array[i]*=2;
  }
  public static void display(double b[]) {
        for(int i=0; i < b.length; i++)
              System.out.print(b[i] + " ");
  }
  public static void main(String[] args) {
        double [] a = {1, 3, 4, 5};
        method(a);
        display(a);
  }
}
```

Answer (Output):

3.4) **(3 marks)** Consider the **Student** class given below which is used to create students who can take a midterm. Complete the `setMidtermScore` method which updates the student's score according to the rules listed below. You are ONLY allowed to use **if-else statements**.

(a) A student who is a genius will always get 100 regardless of midterm's difficulty and student's study.

(b) A student who studies for a midterm will get 100 if the midterm's difficulty rating is less than or equal to 7, otherwise they will get 80.

(c) A student who doesn't study will get 50 if the midterm's difficulty rating is less than 5, otherwise they will get 0.

```
public class Student{
   private int score;
   private boolean genius;
   private String name;
   public Student(String name, boolean genius){
         this.name = name;
         this.genius = genius;
         score = -1; // exam not taken yet
   }
   public int getScore() { return score; }
   public void setMidtermScore(int difficulty, boolean studied){

   ..............................................................
   ..............................................................
   ..............................................................
```

Question 4. (10 Marks)

In this question, you will complete the implementation of a program that allows an airline to assign seats of an airplane to passengers.

4.1) (2 marks) The class **Passenger** given below will represent one passenger by their name and an integer representing an identifier for their family. Passengers from the same family will have the same familyID.

Fill the blank lines in the following implementation of class Passenger

```
public class Passenger {
     private String name;
     private int familyID;

     public void setName( String s ) { name = s;}
     public String getName() { return name; }
     public void setFamilyID( int fid ) {

          ________________________;
     }
     public int getFamilyID() {

          return familyID;
     }
     // If this passenger's family id is greater than other's, return 1
     // If this passenger's family id is less than other's, return -1;
     // If they have the same family, return 0;
     public int compareFamily( Passenger other ) {

          if ( ________________________________)

               ____________________;

          else if(_________________________________)

               ____________________;
          else
               ____________________;
     }
}
```

4.2) (6 marks) To represent the assignment of passengers to seats on a flight, we will use the class **Flight**. You may assume that each flight consists of 20 seats. Each row consists of four seats, and there is an aisle between the second and third seat of each row. To represent the seats, we will use an array of 20 elements. We can number the seats on the flight starting from the first row, left to right, and associate each one seat with one index of the array. For example, the first 20 entries in the array will correspond to the seats as shown:

r0	0	1	2	3
r1	4	5	6	7
r2	8	9	10	11
r3	12	13	14	15
r4	16	17	18	19

and will be represented by a one dimensional array of 20 elements:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

In class **Flight** given below, each entry in our array is going to be a reference to a **Passenger**. If no passenger is booked to this seat, the reference will be **null**.

```
public class Flight{

    private int rows=5;
    private Passenger [] passArray;

    public Flight( Passenger [] p ) {

        passArray = new Passenger[rows * 4];
        for(int i = 0; i < p.length;i++)
            passArray[i] = p[i];
    }

    //This method will sort the array seats in increasing order
    //of familyID with empty seats at the end of the array.
    private static void sortByFamily( Passenger [] seats ) {
        for(int i = 0; i < seats.length;i++)
            for(int j = 0 ; j < seats.length - i - 1; j++) {
                if( seats[j+1] != null ) {
                    if( seats[j] == null
                        || seats[j].compareFamily( seats[j+1] ) > 0 ) {
                        Passenger temp = seats[j];
                        seats[j] = seats[j+1];
                        seats[j+1] = temp;
                    }
                }
            }
    }
    public void print ( ) {
        // You will implement this in Question 4.2.a
    }
    //Reverse part of the array. From index (including) low
    //to index (including) high
    private static void reverse(Passenger [] arr, int low, int high) {
        // You will implement this in Question 4.2.b
    }
    //Rearrange seating so that family members are seated next to or in front of
    //each other.
    public void reArrange() {
        // You will implement this in Question 4.2.c
    }
}
```

4.2.a) Method print: This method will draw the seats of the flight. If a seat is free, it will be represented by an "x". If the seat is booked, then the family ID of the passenger is printed. You may assume that family Ids are one digit numbers (between 0 and 9). For example, this method may print the flight's seating as follows:

```
12 21
31 12
34 5x
xx xx
xx xx
```

Complete the implementation of the method print:

```
public void print ( ) {
    for(int r =0; ________________ ; r++ ) {//For each row
        for(int i =0 ; i < 4 ;i++) {
            if( i == 2 )
                System.out.print(" ");
            if(_____________________________)
                System.out.print("x");
            else
                System.out.print(_____________________________);
        }
        ____________________________;
    }
}
```

4.2.b) Method reverse: This method receives an array, a starting index and an ending index. It reverses only those elements between (including) the starting and ending indices. For example, if it executed with the following arguments. **Suppose arr is:**

0	1	2	3	4	5
Passenger 1	Passenger 2	Passenger 5	Passenger 11	Passenger 13	Passenger 3

With argument low = 1, and argument high = 4, the array becomes:

0	1	2	3	4	5
Passenger 1	Passenger 13	Passenger 11	Passenger 5	Passenger 2	Passenger 3

Complete the implementation of the method reverse:

```
private static void reverse(Passenger [] arr, int low, int high) {
    int mid = (high-low + 1)/2;
    for(int c = 0; ________________ ;c++) {
        Passenger temp = ____________________;
        ____________________________;
        ____________________________;
    }
}
```

4.2.c) Method reArrange: This method will change the seating of passengers so that members of a single family are seated either next to each or behind each other. Since passengers belonging to the same family will have the same family ID, you can use the method **sortByFamily**. However, note that this is not enough. Consider the following seat assignment for 15 passengers to 20 seats:

```
1x 26
31 12
34 54
52 25
xx xx
```

If sorted by family ID, the seating assignment becomes:

```
11 12
22 23
34 45
55 6x
xx xx
```

However, this is clearly not enough, as members of family 3 seated far away from each other, and members of family 2 are also far away from each other. Your task is to use sorting and the method **reverse** to correctly rearrange the seating assignment so that family members are seated as close as possible to each other. For example, you may assign seats as follows, so that members of families two and three are seated next to their families.

```
11 12
32 22
34 45
x6 55
xx xx
```

Complete the implementation of the method reArrange (hint: use the methods sortByFamily and reverse):

```
public void reArrange() {
    sortByFamily(_________________);

    for(int r = 0; _________________ ; r++)
        if(_________________)//reverse the odd rows
            _____________________________________________;
}
```

4.3) (2 marks) Fill in the blanks of the following program. This program will read passenger information from the user, put those passengers into seats, arrange the seating assignment so that families are seated together and then display the airplane's seats on the screen.

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

    public static void main ( String [] args ) {
        Scanner in = new Scanner(System.in);
        int passCount;

        System.out.println("How many passengers?");
        passCount = in.nextInt();
        Passenger [] temp = new Passenger[passCount];

        // Filling the temp array with passenger information.
        for(int i = 0; i < passCount;i++ ) {
            temp[i] = ____________________________;
            System.out.println("Enter Family ID:");
            ____________________________________;

            in.nextLine();
            System.out.print("Enter Passenger's Name:");
            ____________________________________;
        }

        // Create a new flight.
        ____________________________________;

        //Put families together.
        ____________________________________;

        //Display the airplane's seats on the screen.
        ____________________________________;
    }
}
```