BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY, DHAKA

L-1/T-2 B. Sc. Engineering Examinations (January 2020 Term)

Sub: CSE 107 (Object Oriented Programming Language)

Full Marks: 180 Section Marks: 90 Time: 2 Hours (Sections A + B)

USE SEPARATE SCRIPTS FOR EACH SECTION

The figures in the margin indicate full marks.

SECTION – B

There are FOUR questions in this section. Answer any THREE.

All the questions in this section are related with Java programming language

5(a). What is the difference between the following two declarations in Java? (10)

i. int c [ ][ ], x
ii. int [ ][ ] c, x

Write 2 (two) different ways of creating the following array in Java.

0			
1	2		
3	4	5	
6	7	8	9

5(b). What are the problems with the following Java code? (10)

```
public class TestStatic {
        static int a = 3, b;
        int c;

        {
                c = 10;
        }

        static {
                b = a*4;
                c = b;
        }

        int f2() {
                return a*b;
        }

        static void f1(int x) {
                System.out.println("x = " + x);
                System.out.println("a = " + a);
                System.out.println("b = " + b);
                System.out.println("c = " + c);
        }

        public static void main (String [] args) {
                f1(42);
                System.out.println("b = " + b);
                System.out.println("Area = " + f2());
        }

}
```

5(c). Consider the following code segment: (10)

```
public class Main {
        public static void main (String [] args) {
        int a = minmax ("min", 2, 1, 6, 4, 5); // a = 1
        int b = minmax ("min", 3, 0, 6); // b = 0
        int c = minmax ("max", 1, 2, 6, 5); // c = 6
        int d = minmax ("max", 1, 3, 7); // d = 7
   }
}
```

Write the above minmax function in Java. You are allowed to write only one minmax function.

6(a). Consider the following code segment: (10)

```
interface i1 {
        default void f1() {
        }
        static void f2() {
        }
        void f3();
}
interface i2 {
        void f4();
        void f5();
}
```

```
abstract class c1 implements i1 {
        abstract void f6();
        final void f7() {
        }
}
class c2 extends c1 implements i2 {
        // your code
}
```

Write minimum code in class c2 for successful compilation. You can't define c2 as abstract.

6(b). Suppose there are 8 methods defined as follows: (10)

void m1(), void m2(), void m3(), void m4(), void m5(), void m6(), void m7(), and void m8()

There are also 4 interfaces named as: interface x1, x2, x3 and x4.

There is also a class named MyClass that needs to be forced to implement all the above 8

methods, where you have to maintain the following constraints:

i. Each interface can define at most 2 methods.
ii. The class MyClass can only implement 1 interface.

Write Java code for MyClass to achieve the above scenario.

6(c). Consider the following code segment: (10)

```
public class Institute {
   private int eiin;
   private int shift;
   private int version;
   private int group;

   public Institute (int eiin, int shift,
         int version, int group) {
      this.eiin = eiin;
      this.shift = shift;
      this.version = version;
      this.group = group;
   }
```

```
public static void main (String [] args) {
      Institute i1 = new Institute (135790, 1, 1, 0);
      Institute i2 = new Institute (135790, 1, 1, 0);
      System.out.println(i1.equals(i2));
      HashMap map = new HashMap();
      map.put (i1, 100);
      System.out.println(m.get(i2));
   }
}
```

The expected output of the above code:
true
100

Complete the Institute class to achieve the expected output.

7(a). With Java threads, it is very easy to parallelize computations. Suppose you are in a job interview and the interviewer asks you to write Java code to find out summation of 1 to 10000000. You can't use any simple equation; you can only use loops. But you are asked to divide the work equally among 10 different threads. Write complete Java code to compute the summation of 1 to 10000000 by dividing the work equally among 10 different threads. The main thread will wait for the 10 threads to finish and will only print the final summation. (10)

7(b). Consider two entities, the Writer and the Reader, who share a common buffer that is a single character. The Writer's job is to generate English letters circularly from A to Z. The Reader's job is to read and display them in the output. The Writer can't write new letter if the Reader does not read the already written letter. The Reader can't read if the Writer does not write any new letter. Write a java code to solve the above-mentioned problem using the concept of inter thread communication. You can use wait/notifyAll or ArrayBlockingQueue. (10)

7(c). Write three different ways to create Threads in Java with short code examples. What is the difference between synchronized method and synchronized statement? (10)

8(a). What do you mean by auto-boxing and auto-unboxing? Explain with code examples. When you shouldn't use them? Write three differences between Hashtable and HashMap? (10)

8(b). Write a generic interface named iQueue with methods enqueue, dequeue and isEmpty. Then write a generic class Queue that implements the iQueue interface. Please note that iQueue interface only supports numeric types. (10)

8(c). Consider the following class: (10)

```
class Product {
        private String name;
        private double price;

        Product (String name, double price) {
                this.name = name;
                this.price = price;
        }
        public String getName () {
                return this.name;
        }

        public double getPrice () {
                return this.price;
        }
}
```

Write Java code for the following:

i. Define an ArrayList named myProducts that can store a list of Product.
ii. Generate 4 random Product with names 'A' to 'D' and random price and add them to myProducts.
iii. Sort myProducts based on Product's name in ascending order. You can't use your own sorting techniques. You can change the Product class if necessary.