United International University
Department of Computer Science and Engineering
CSI 211 Object Oriented Programming
Final Exam, Fall 2016
Total Marks: 40, Time: 120 minutes

Answer any 5 questions ($5 \times 8 = 40$)**.**

1. Find out if the following JAVA programs have any error. List the errors if any. Fix the code and rewrite after the list. You cannot delete any line of code. However, you are allowed to edit or add any code as per requirement. [2+3+3]

(a)
```
class Super {
    void show() throws ArithmeticException {
        System.out.println("parent class");
    }
}
public class Child extends Super {
    void show() throws Exception {
        System.out.println("child class");
    }
    public static void main(String[] args) {
        try {
            Super s = new Child();
            s.show();
        } catch (Exception e) {
        } catch (ArithmeticException e) {
        }
    }
}
```

(b)
```
public class ServerWithError {
    public static void main(String[] args) {
        try {
            Socket cs = ss.accept();
            ServerSocket ss = new ServerSocket("localhost", 1311);
            OutputStreamWriter osw = new OutputStreamWriter(cs);
            osw.write("hello");
        } catch (NullPointerException e) {
        }
    }
}
```

(c)
```
public class FileWithError {
    public static void main(String args[]) throws IOException {
        try {
            FileReader in = new FileReader("input.txt");
            FileWriter out = new FileWriter("output.txt");

            String c;
            while ((c = in.readLine()) != null) {
                out.writeLine(c);
            }
        } finally {
            in.close();
            out.close();
        }
    }
}
```

2. Write a JAVA program which will read from three (3) files: **firstnames.txt**, **middlenames.txt**, **lastnames.txt**. All these files have equal number of lines, each line in these files represent the *first name*, *middle name* and the *last name* of a person. Now, your code should create a new file named **fullnames.txt** and write the full names of the persons in the following format: *first name*<space>*initials of middle name*<space>*last name*. Three input files may look like as shown below: [8]

   Your code should generate the output file in the following manner:

3. Write a JAVA program (**Application.java**) which takes four (4) inputs from users (keyboard): *name*, *age*, *cell phone number* and *email address*; writes these information in one file (**output.txt**). However, your program should throw the following custom made exceptions if respective conditions fail: **EmptyNameException** if name is empty, **UnderAgeException** if age is less than 18, **InvalidCellException** if cell phone number is less that 11 digits and does not start with a 0 (zero), and **InvalidEmailException** if email address does not maintain the format: *<string>@<string>.<string>*.
   Note: You need to write these four exception classes as well and throw exceptions properly from Application class. However, you don't need to catch these exceptions anywhere. [8]

4. (a) Write a simple counter application which user interface is given bellow.

   In this application, counter value will appear in the text box. When a user clicks the up button the counter value will be increased, and the value will be decreased when the user will press the down button. If the user press reset button, then the counter value in the text box will be set to zero. Importantly, you have to give some necessary conditions so that neither the counter will have a negative value nor a value greater than 100. If this terminal condition occurs, you have to give some warning to the user via showing a message in a label or using a dialog. [5]

   (b) Consider the following program, which was written to sort a list of book according to their price(from higher to lower price). There is a problem in the following code, which you need to resolve. However, you are only allowed to edit the Book class to solve the problem. [3]

```
public class BookSolution {
    public static void main(String[] args) {
        ArrayList<Book> books = new ArrayList<>();
        books.add(new Book("Competitive Programming 101", 180));
        books.add(new Book("Head First Java", 80));
        books.add(new Book("Android App Development", 110));

        Collections.sort(books);
        for(Book book : books){
            System.out.println(book.name+", Price"+book.price);
        }
    }
}
```

```
class Book{
    public String name;
    public double price;

    public Book(String name, double price) {
        this.name = name;
        this.price = price;
    }
}
```

5. Consider a server program, which is running on a computer and can read a list of student name with their CGPA whenever a client connects to the server. After that server send the name and CGPA of a student to that particular client who has the highest CGPA in that student list. Please note that this server program can handle multiple clients; however, it is not necessary that server has to handle these clients parallelly. Unfortunately, a group of hackers, named BABU007, attack the server and erase the whole server program. Fortunately, we recover the client program which is given bellow. Now, as our lead developer is on vacation, we are seeking for your help to write down the server program by carefully observing the client program. If your server program can handle multiple clients properly then your teacher will give eight marks; however, if you can handle single client only, then you will get five marks as rewards. [8]

```
public class Client {
    public static void main(String[] args) {
        try {
            Socket clientSocket = new Socket("127.0.0.1",9999);
            PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true);
            BufferedReader in = new BufferedReader(
            new InputStreamReader( clientSocket.getInputStream()));

            String[] studentName = new String[]{"Anik", "Safat", "Nayeem", "Swapnil", "Arif"};
            Double[] studentCgpa = new Double[]{3.87, 3.75, 3.92, 3.65, 3.89};

            for(int i=0;i<studentName.length;i++){
                out.println(studentName[i]);
                out.println(studentCgpa[i]);
            }
            out.println("end");

            Double highestCgpa = Double.parseDouble(in.readLine());
            String highestCgpaStudentName = in.readLine();

            System.out.println(highestCgpa);
            System.out.println(highestCgpaStudentName);

        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

6. (a) Consider the following program, which is initialized a Student type GenQueue object.

```
public class GenericSolution {
    public static void main(String[] args) {
        GenQueue<Student> studentQueue = new GenQueue<>();

        studentQueue.enqueue(new Student("Imran", 3.92));
        studentQueue.enqueue(new Student("Amee", 3.88));

        while(studentQueue.hasItems()){
            Student st = studentQueue.dequeue();
            System.out.println(st.name+" "+st.cgpa);
        }
    }
}
class Student{
    public String name;
    public double cgpa;

    public Student(String name, double cgpa) {
        this.name = name;
        this.cgpa = cgpa;
    }
}
```

In this generalized queue we can enqueue and dequeue object which type is determined during initialization. Now, unfortunately, we lost the GenQueue class due to the server hacks. Now you have to write a GenQueue class which must have three methods enqueue, dequeue and hasItems. Carefully observe the above code to determine the return type, and the parameter of these methods. Please note that the GenQueue must be a generic class so that it can handle any type of object. [5]

(b) Please complete the following incomplete program. [3]

```
public class ProblemSolution {
    public int num;

    public ProblemSolution(int num) {
        this.num = num;
    }
    public void outerFun(){
        //You have to print the name of Inner
        //You must call innerFun method of Inner in someway
    }
    public class Inner{
        public String name;

        public Inner(String name) {
            this.name = name;
        }
        public void innerFun(){
            System.out.println(num);
        }
    }

    public static void main(String[] args) {
        //You have to call the innerFun method of Inner in someway
        }
    }
```

7. (a) Write a program in JAVA where two different threads will execute in parallel and print even and odd numbers respectively ranging from 1 to 100. Note that synchronization between these two threads are not necessary. Your program may generate scrambled output like: 1 2 4 6 3 5 7 9 11 13 8 15 ... [3]

   (b) Now, rewrite the same program of 7(a) to synchronize the two threads so that they print the numbers in correct order: 1 2 3 4 5 6 ... 99 100. [3]

   (c) Rewrite the program of 7(b) such that the thread which prints even numbers will print two even numbers followed by one odd number. Thus output of your program should be: 1 2 4 3 6 8 5 10 12 7 ... 97 99. [2]