BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
FIRST SEMESTER, 2018-2019
COMPREHENSIVE EXAM (CLOSED BOOK)
PART I - OBJECT ORIENTED PROGRAMMING (CS F213)

DATE: 12th December 2018 **MAX MARKS: 22.5** **TIME: 60 Min**

Important Instructions:

1. The exam comprises of Part I, Part II and Part III. Part I and II are closed book and Part III is open book. The maximum duration for Part I is 1 hr. You can receive the Part III question paper once you have submitted the Part II.
2. **Attempt all questions. Each question carries 0.75 mark.**
3. **More than one option may be correct. Choose all the correct option(s) to get any credit.**
4. **For every incorrect answer 0.25 marks will be deducted.**
5. **Overwriting and cutting is not acceptable and will not be evaluated.**

Q.1. Which of the following characteristics of an object-oriented programming language restricts behavior so that an object can only perform actions that are defined for its class?

a. Dynamic Binding
b. Polymorphism
c. Inheritance
d. Encapsulation

Q.2. When would you use a private constructor?

a. When you get bored with public
b. If you want to disallow instantiation of that class from outside that class
c. If you want to protect your class's members from outside modification
d. Never, it's not allowed

Q.3. What is the output when you try to compile and run the following code?

```
public class Switch {
        public static void main(String[] args) {
                int i = 1;
                switch( i ) {
                        case 0:
                                int j = 0;
                                System.out.print( j );
                        case 1:
                                int j = 1;
                                System.out.print( j );
                        case 2:
                                int j = 2;
                                System.out.print( j );
                        default:
                                int j = -1;
                                System.out.print( j );          }       }
}
```

a. 12-1
b. 1
c. 12
d. The code does not compile

Q.4. A class which implements the ActionListener interface must implement which method?

a. void handle(ActionEvent e)
b. void actionPerformed(ActionEvent e)
c. void eventDispatched(AWTEvent e)
d. String getActionCommand(ActionEvent e)

Q.5. What is the most specific result of the following code?

```
Integer[] someInts = new Integer[100];
int sum = 0;
for ( Integer i : someInts )
        sum += i;
System.out.println( sum / someInts.length );
```

a. 0
b. The sum of 100 integers
c. NullPointerException
d. The code will not compile

Q.6. Given the following method and class signatures:

```
public class A extends Exception {...}
public class B extends A {...}
public class C extends B {...}
public void doStuff() throws A,B,C
```

The following code does not compile. Why?

```
try {
        doStuff();
} catch(A a) {
        a.printStackTrace();
} catch(B b) {
        b.printStackTrace();
} catch(C c) {
        c.printStackTrace();
} finally {
        System.out.println("I love exceptions!");
}
```

a. The catch blocks for exceptions of type B and C are unreachable.
b. A finally block cannot be used with multiple catch blocks.
c. B and C are not exception classes since they do not extend class Exception and therefore cannot be caught.
d. No one loves exceptions and therefore the finally block fails to compile.

Q.7. What is the output of the following program?

```
public class A {
public static int doStuff(double x, double y) {
return (int)(x/y);}

public static void main() {
float x = 6.0;
int y = 11;
x = A.doStuff(y,x);
System.out.print("x="+x+", y="+y);}
}
```

a. x=1, y=11
b. this program does not compile
c. x=6.0, y=11
d. x=1.0, y=11

Q.8. Let a priority of a parent thread be 3 and three child threads are spawned from the parent thread. What will be the priority of the three child threads?

a. 3,4,5
b. 5,5,5
c. 3,3,3
d. 0,1,2

int = (int)(19L)
long = int
float = int

Q.9. An array object, ArrayOne, is created as:

float [][] ArrayOne;
ArrayOne = new float[20][10];
Suppose ArrayOne is passed as an argument to a method in which the corresponding parameter is named someArray. What should the declaration of someArray look like in the parameter list of the method?

a. float [][] someArray
b. float someArray[]
c. float [] someArray[20]
d. float someArray[20][10]

Q.10. When writing data to a file using a FileOutputStream, at what point is the data actually written to the file?

I. Immediately after the write function is called
II. When the data buffer is full
III. When the close function is called

a. I only
b. III only
c. II and III
d. II only

Q.11. Suppose the class Undergraduate extends the class Student which extends the class Person.

Given the following variable declaration:
Person p = new Person();
Student s = new Student();
Undergraduate ug = new Undergraduate();
Which of the following assignments are legal?
I. p = ug;
II. p = new Undergraduate();
III. ug = new Student();
IV. ug = p;
V. s = new Person();

a. III and IV
b. I and IV
c. I and II
d. II, III and V

Q.12. Which of the following are true regarding the use of generics and parameterized types in Java?

I. Generics provide type safety by shifting more type checking responsibilities to the compiler.
II. Generics and parameterized types eliminate the need for downcasts when using Java Collections.
III. When designing your own collections class (say, a linked list), generics and parameterized types allow you to achieve type safety with just a single class definition as opposed to defining multiple classes.

a. I and II
b. II and III
c. I, II, and III
d. I and III

Q.13. Which of the following statements are false?

a. Static methods cannot be synchronized
b. If a class has synchronized code, multiple threads can still access the non-synchronized code
c. Variables can be protected from concurrent access by using synchronized keyword.
d. When a thread sleeps, it releases the locks.

Q.14. Given the following definitions, which assignments are legal?

```
class Box<T>{}
class SuperBox<T> extends Box<T>{}
I. Box<Object> b = new Box<String>();
II. Box<String> b = new SuperBox<String>();
III Box<Object> b = new SuperBox<String>();
```

a. I, II, III
b. II only
c. I and III only
d. I only

Q.15. What is the output of a call to the printNums() method? Assume inFile has been properly linked to the file whose content is shown below and outFile has been properly linked to some output file.

```
public static void printNums() {
        int n;
        String line;
        line = inFile.readLine();
        if (line != null) { //If not EOF ..
                n = Integer.valueOf(line).intValue();
                outFile.print(n + " ");
                printNums();
                outFile.print(n + " ");
        }
}
```

15
23
21
19

a. 15 23 21 19 19 21 23 15
b. 19 21 23 15 19 21 23 15
c. 15 23 21 19
d. 19 21 23 15

Q.16. What is the number of threads in the given program?

```
public class ThreadExtended extends Thread {
        public void run() {
                System.out.println("\nThread is running now\n");
        }
        public static void main(String[] args) {
                ThreadExtended threadE = new ThreadExtended();
                threadE.start();
        }
}
```

a. 0
b. 1
c. 2
d. 3

Q.17. Which of the following statements are true?

a. Thread will resume execution as soon as its sleep duration expires.
b. Synchronization can prevent two objects from accessing the same thread
c. wait() method is overloaded to accept a duration.
d. Both wait() and notify() must be called from a synchronized context.

Q.18. Which of the following class level (nonlocal) variable declarations will not compile?

a. protected int a;
b. transient int b = 3;
c. private synchronized int e;
d. volatile int d;

Q.19. What will be the output of the following code?

```
public class WaitTest {
    void method1(String[] args)    {
        synchronized(args)    {
            System.out.print("2 ");
            try                                    {
            wait();
            }
            catch(InterruptedException e){ }
        }
    }
    public static void main(String [] args)    {
        System.out.print("1 ");
        new WaitTest().method1(args);
        System.out.print("3 ");
    }
}
```

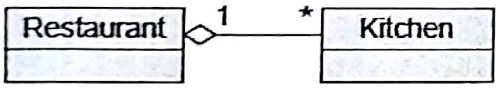
a. Fails to compile because the IllegalMonitorStateException of wait()
b. 1 2
c. 1 3
d. 1 2 3

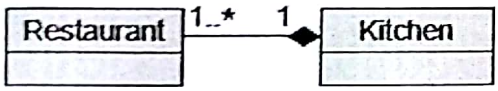
Q.20. What is the output of the following?

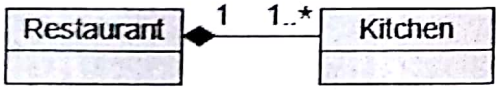
```
final int i;
i = 20;
System.out.println(i);
```

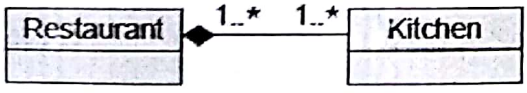
a. 20
b. Compiler Error
c. 0
d. Garbage Value

Q.21. How would you model the following situation using UML class diagram: Every restaurant has at least one kitchen, one kitchen is part of exactly one restaurant?

a.


b.


c.


d.


Q.22. Which of the following is not the use of "this" keyword?

a. this.memberName
b. this(args)
c. this.methodName(args);
d. this{statements;}

Q.23. What will be the output of the following?

```
HashSet<String> set=new HashSet<String>();
set.add(null);
set.add("One");
for(String s: set)
  System.out.println(s);
```

a. NullPointerException
b. null, One
c. One

Q.24. Complete the following code.

```
public class Output {
    public static void main(String args[])
    {
      int a =5;
      //insert the missing code here
       System.out.println(s);      }
  }
```

a. String s = Integer.toString(a)
b. String s = String.valueOf(a)
c. String s = new String(a);
d. None of the above

Q.25. What is the output of the following code?

```
class Vehicle
{
   int id = 120;
   public int getId()
   {
      return id;   }
}

public class Car extends Vehicle
{
   int id = 100;
   public int getId()
   {
      return(id - 20);
   }
   public static void main(String[] args)
   {
      Vehicle vc = new Car();
      Car car = (Car) vc;
      System.out.print(vc.id + ", " + vc.getId() + ", ");
      System.out.print(car.id + ", " + car.getId());
   }
}
```

a. 120, 100, 120, 100
b. 120, 80,120, 80
c. 120, 80, 100, 80
d. 100, 80, 120, 80

Q.26. Predict the output

```
class Parent{
   int a;
}

public class Child extends Parent{
 public static void main(String args[]){
  Child c=null;
  System.out.println(c instanceof Parent);
 }
}
```

a. True
b. False

Q.27. Will the following code compile?

```
List<Number> list1 = null;
List<? super Integer> list2 = null;
list1 = list2;
```

a. Yes
b. No

Q.28. Given the following definitions, which assignments are legal?

```
class Box<T>{}
class SuperBox<T> extends Box<T>{}
```

a. Box<Object> b = new Box<String>();
b. Box<String> b = new Box<String>();
c. Box<Object> b = new SuperBox<String>();
d. Box<String> b = new SuperBox<String>();

Q.29. Will the following code compile?

```
public class test{
   public static void main(String[] args) {
     List<Float> list1= Arrays.asList(1f,2f,3f);
     List<Number> list2=Arrays.asList(1.1,2.2,3.3);
     List<Double> list3=Arrays.asList(1.1,2.2,3.3);
     printlist(list1);
     printlist(list2);
     printlist(list3);
   }
    private static void printlist(List<? super Float> list)     {
       System.out.println(list);
   }
}
```

a. Yes
b. Compilation error in list2
c. Compilation error in list3
d. Compilation error ins list2 and list3

Q.30. What will be the output of the following code?

```
interface A {
   static void methodA(){
      System.out.println("Inside A");
   }
}

interface B {
   default void methodA(){
      System.out.println("Inside B");
   }
}

public class Test implements A, B{

   public static void main(String []args){

      Test t = new Test();
      t.methodA();
   }
}
```

a. Inside A
b. Inside B
c. Compilation Error

***** BEST OF LUCK *****

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
FIRST SEMESTER, 2018-2019
COMPREHENSIVE EXAM (CLOSED BOOK)
PART II - OBJECT ORIENTED PROGRAMMING (CS F213)

DATE: 12th December 2018 | MAX MARKS: 27.5 | TIME: 60 Min

Important Instructions:

1. The exam comprises of Part I, Part II and Part III. Part I and II are closed book and Part III is open book. The maximum duration for Part I is 1 hr. You can receive the Part III question paper once you have submitted the Part II.
2. **Attempt each question on a fresh page. Subdivisions in the question can be answered continuously.**
3. **Assume that all the necessary packages and classes are already imported in your program.**
4. **Read the all the subdivisions carefully before you start to answer a question.**

Q1. Let there be one Moderator and 'n' Players. The moderator displays 10 random numbers (between 0 – 50) on a display screen with one-minute gap between the numbers. Each player is given a card containing 10 random numbers (between 0 - 50). As the number is displayed, the player strikes the number on his card if it matches with the one on the screen. The player who strikes three numbers first will be announced as a winner and the moderator stops generating numbers if a player wins before all the 10 numbers are generated. The numbers generated by the moderator and the player can be redundant. For eg. If the moderator generates a 20 twice and if a player has one 20. After the player strikes the number, it should not be considered as a match, when 20 appears again on the display screen.

The moderator generates 10 numbers and stores it in an array list of integers. When the moderator is generating the number, the players should not read the array list. The moderator should generate the next number only after all the players have read the previously generated number. The array list should not be read by two players at the same time. The snippet of the above mentioned task is given, write codes to complete the missing parts.

Note: Do not modify the variable names or add class/local variables.

```
class Message {

   ArrayList<Integer>screen=new ArrayList<Integer>();   //used to store the numbers to be displayed
   int count = 0;                                        //used to track the no. of players who have read a particular number
   int noOfPlayers;                                      //Total no. of players
   boolean available=false;                              //Availability flag
   int value;                                            //used to store the nu
   String winner ="No one";                              //used to store the name of the winner

public int getMsg(int pos) {

/* Q1.a) Write the code to read the number from the screen and return it to the player; this module also keeps track
of the number of players who have read a particular number generated by the moderator. */                    [1 Mark]
}

public void setMsg(int value) {
     screen.add(value);       }

public void won(String name)      {
   winner =name;      }
}

class Player implements Runnable{

   private Message msg;
   private int match=0;

   public Player(Message m){
      this.msg=m;
```

```
    }

    public void run() {
       String name;                                   //used to store the name of the player
       int token;                                     // used to store the number displayed on the screen
       int i = 0;                                     //used for looping

       ArrayList<Integer> card =new ArrayList<Integer>();
```

/*Q1.b) Write a code to retrieve the name of the current player thread and to generate 10 random numbers for the players' card*/ [1 Mark]

```
       System.out.println(name+" tokens are: "+card);
       i =0;
```

while(/* Q1.c) Write the condition so that player thread plays for 10 rounds or until someone wins*/) [1 Mark]

```
       {
```

/*Q1.d) Write the code to do the following. Let the player thread be synchronized on the Message object, it waits until the moderator generates the number and sets the availability flag to true, it also checks if the player is not trying to access beyond the size of the array list. When the data is available, it retrieves the i^{th} number from the screen and stores it in the variable 'token'. The availability flag is set to false if all the players have read a particular number generated by the moderator and notifies the waiting threads. The 'count' variable in the Message class is reset to zero, if all the players have accessed the number generated by the moderator. The token is compared with the number on the players' card, if it equal, the match counter is incremented and is removed from the array list. Thus, the player can strike off matching numbers. If the match counter is three, assign the name of player as a winner using the method won(). This process is repeated for all the numbers displayed on the screen. */ [8 Marks]

```
                i++;
       }
       System.out.println("The no. of matches for " + name + " is: "+ match);
    }
}

class Moderator implements Runnable {

   private Message msg;
   int number;

   public Moderator(Message msg) {
      this.msg = msg;
   }

   public void run() {
      String name;
      int i=0;
```

/*Q1.e) Write the code to retrieve the name of the moderator thread */ [0.5 Marks]

```
       System.out.println(name+" started");
```

while(/* Q1.f) Write the condition such that the moderator generates numbers for 10 rounds or until someone wins*/) [1 Mark]

```
       {
```

/* Q1.g) Write the code to generate a random number for the i^{th} round*/ [0.5 Marks]

```
          System.out.println(name+" Generated:"+number);
```

/* Q1.h) Write the code to do the following. Let the moderator thread be synchronized on the Message object. It waits until all the players have read the data, by checking the availability flag and the variable 'count'. Once it becomes ready, it sends the generated number to the screen using the method setMsg(). Let the moderator sleep for 1 minute and it sets the availability flag to true. All the waiting threads are then notified. */ [4 Marks]

```
                i++;
        }
    }
}
public class Casino {
        public static void main(String[] args) throws InterruptedException {
                Message msg = new Message();
                msg.noOfPlayers = 2;

                Thread Moderator, Player1, Player2;
```

/Q1.i) Write the code to instantiate the Moderator and two Player threads. Name the threads as Moderator, Player1 and Player2. Start all the threads and let the monitor thread wait for all the threads to join */ [3 Marks]

```
            System.out.println("The winner is :" + msg.winner);          }
}
```

Q2. Let an application allows its users to create an account by setting their gmail ID as the username and it also enforces that the user should provide a password that contains atleast one Capital letter. Let the Email class contains the following members: String username, String password, Boolean success.

Q2. a) Let the setUser() method obtains a string input and checks if the username is a gmail ID, if true, it sets the username else it throws a CheckFormatException. [1 Mark]

Q2. b) The setPass() method checks if the password contains atleast one capital letter else it throws a CheckFormatException. Use, if and else conditions only. [1 Mark]

Q2. c) Let CheckFormatException be an inner class in the Email class and it is a run time exception. The String message in the inner class can be used to describe the condition causing the exception. [1 Mark]

Let the Person class contains the person name and Email object as its data members. It contains 2 static HashMap as its members, the first HashMap (dB1) is used to map the name with the username and the second HashMap is used to map the username with the password.

Q2. d) Write the code for the createAccount(), it receives the First name, Last name, username and password as input. It appends the First name and Last name and stores it as the person's name. This method checks if the name is already found in dB1, if found, it raises a CheckFormatException, else, it adds the records to dB1 and dB2. [2 Marks]

Q2.e) Write the code to instantiate the scanner class to read input from the command prompt. Read the no. of accounts to be created from the command prompt and create an array of Person references. [1.5 Marks]

Q2.f) Write the code to create Account for the specified no. of Persons. Obtain the First name, Last name, email id and password from the command prompt. [1 Mark]

Note: CheckFormatException object if printed, displays the following strings

1. **Data already exists – if the name is already found in dB1**
2. **Not a Gmail ID – if the email id provided is not a gmail ID**
3. **Use atleast one capital letter in password – if the password doesnot contain atleast one capital letter**

```
class Email {

        String username;
        String password;
        Boolean success =false;

        void setUser(String username) { //Write your code for Q2.a  }
```

```
        void setPass(String password) { //Write your code for Q2.b }

        //Write your code for Q2.c }

class Person{
        private String name;
        Email e =new Email();
        private static HashMap<String,String> dB1 = new HashMap<String,String>();
        private static HashMap<String,String> dB2 = new HashMap<String,String>();

        public void createAccount(String fname, String lname, String username, String password)
        { // Write the code for Q2.d  }

        public String toString()
         {
                if(e.success)
                        return "Email Id for "+name+" is successfully created";
                else
                        return "Exception Generated";
         }

        public void displayDB()
        {
                System.out.println(dB1);
                System.out.println(dB2);
        }

}

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

                Scanner in;
                int noOfPersons;
                int i;

                System.out.println("Enter the number of Accounts to be created");
                //Write the code for Q2.e

                System.out.println("Enter the first name, last name, email id and password");
                for(i =0;i<noOfPersons;i++)
                {
                P[i]=new Person();

                try { //Write the code for Q2.f }
                catch(Exception E) {
                        System.out.println(E);
                }
                System.out.println(P[i]);
                }
                in.close();
                }
}
```

***** BEST OF LUCK *****

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
FIRST SEMESTER, 2018-2019, COMPREHENSIVE EXAM (OPEN BOOK TEST)
PART III - OBJECT ORIENTED PROGRAMMING (CS F213)

DATE: 12th December 2018 MAX MARKS: 30 TIME: 60 Min

Instructions for Q.1 (a) to Q.1 (c) – Write the name of the software design pattern that you think can be appropriately applied in a given context and then write your justification for that in no more than 200 words. [IMPORTANT: You will lose marks for being verbose]

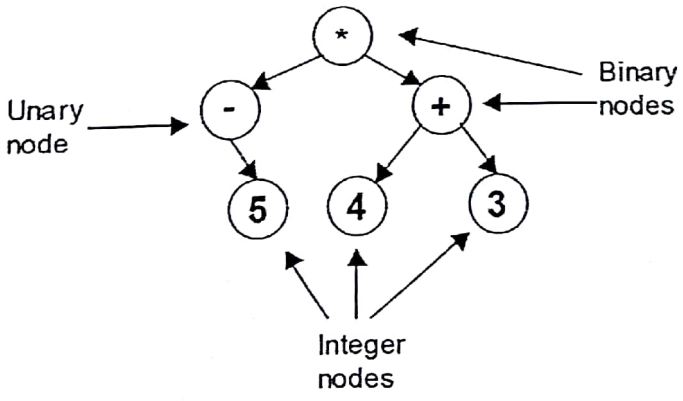
Q.1 (a) A server receives messages from a client containing operations that the server should do. The server is supposed to execute these operations in the order received. **What design pattern should be used in the server's implementation to accomplish this?**

Q.1 (b) You want to add the ability to write text to **Java standard output** [using `System.out.println()`] or a **file** in all UPPERCASE. You want this to work with client code that can write text to a PrintStream, but it writes the text in all LOWERCASE and requires conversion. **What pattern should you use and why?**

Q.1 (c) Suppose you are implementing some application (say **XImpl**) for an android mobile phone and you have created an object to keep track of whether the phone is in airplane mode. There are many other applications that need to know when the phone goes into airplane mode so that they can start running in low-power mode to preserve your battery life and when it comes out start running normally, but you do not know exactly what those applications are. **What pattern should you use and why? Draw a UML class diagram by choosing appropriate names for the classes for XImpl application.**

[2 + 2 + 6 = 10 Marks]

Q.2. Expression tree consist of nodes containing operators and operands. Operators have different precedence levels; e.g., multiplication takes precedence over addition. The multiplication operator has two arguments, whereas unary minus operator has only one. Operands are integers, doubles, variables, etc. We will deal with integers in this question. Expression tree may be "evaluated" via different traversals, e.g., in-order, post-order, pre-order, level-order. The evaluation step may perform various operations, e.g., traverse and print the expression tree, return the value of the expression tree. Consider the example of the expression tree given below:



Show how Abstract Factory could be effectively used in this application. Firstly, show the UML class diagram. Use proper class names and some of the important operations and instance variables required for use. Secondly, show some code snippets in java for each class in your design and illustrate how the nodes would be created. **[20 Marks]**

***** BEST OF LUCK *****