Examination	:	**First Sessional**	**Seat No.**	:	
Date	:	**03/08/2017**	**Day**	: **Thursday**	
Time	:	**11:30 to 12:45 PM**	**Max. Marks**	: **36**	

INSTRUCTIONS:
1. Figures to the right indicate maximum marks for that question.
2. The symbols used carry their usual meanings.
3. Assume suitable data, if required & mention them clearly.
4. Draw neat sketches wherever necessary.

Q.1 **Do as directed**. **[12]**

(a) 1. If we write, Account [] as=new Account[10]; [2]
How many times constructor will be called? Give reason for your answer.
2. What will be the output of the following program code?
```
package mypack.test;
public class Main {
        public static void main (String args[] ) {
                Main object = new Main();
                System.out.println(object.toString());
        }
}
```

(b) 1. If "a" and "b" are object references, then what is the difference between "a == b" and [2]
"a.equals(b)"?
2. What will be the output of the following code?
```
class Test{
  public static void main(String[] args){
        System.out.println(1+2+"=1+2="+1+2);
  }
}
```

(c) State whether any error exists in the following code fragment. If so, correct the error [2]
and give output.
```
        public static void main(String s[]) {
                String str1 = new String("ddu");
                String str2 = new String(str1);
                str1.concat(" nadiad");
                System.out.println(str1 == str2);
                System.out.println(str1);
        }
```

(d) Write the output of the following code [2]
```
class CallingSequenceBase{
  static int si=initializeStaticVar();
  int ii=initializeInstanceVar();
  private static int initializeStaticVar(){
        System.out.println("> CallingSequenceBase.initializeStaticVar()");  return 1;
  }
  private int initializeInstanceVar(){
        System.out.println("> CallingSequenceBase.initializeInstanceVar()");      return 10;
  }
  CallingSequenceBase(){
        System.out.println("> CallingSequenceBase()");
  }
}
class CallingSequence extends CallingSequenceBase{
  static int sj=initializeStaticVar();
  int ij=initializeInstanceVar();
  private static int initializeStaticVar(){
        System.out.println("> CallingSeqeunce.initializeStaticVar()");  return 1;
  }
  private int initializeInstanceVar(){
        System.out.println("> CallingSeqeunce.initializeInstanceVar()");  return 20;
  }
  CallingSequence(){
        System.out.println("> CallingSequence()");
  }
  public static void main(String[] args){
        CallingSequence obj=new CallingSequence();
  }
}
```

(e) If we make constructors as private, then how can we allow creating instances of such class? [2]
Write code fragment for above.

(f) State True/False [2]
(i) A final class can have instances.
(ii) Static members cannot be accessed from non-static members.
(iii) A class cannot implement more than one interface.
(iv) An interface can extend from more than one interface.

Q.2 Attempt *Any Two* from the following questions. **[12]**

(a) 1. Write a java program to read 6 numbers, separated by white space, from command line. Store [3]
the elements in a dynamic size array, as shown below.

2. Write code fragments to perform the following operations on a given sting [3]
str ="dharmsinh desai university".
a. Find the length of string
b. Replace the character 'd' by 'n'
c. Convert all characters in uppercase
Extract and print "dharmsinh" from given string.

(b) Implement a command line arithmetic calculator that allows the following operations: +, –, *, [6]
and /, which represent addition, subtraction, multiplication, and division, respectively, on
integer numbers. Operands and operator are passed via command line as a single argument. For
example, if we want to do addition of 20 and 30, we should execute the program using the
following syntax.
$java Calc 20+30
There is no space between 20 and +; and between + and 30. (Hint: Use StringTokenizer)

(c) Answer the following: [6]
1. Explain Garbage collection using reference count mechanism in java using example.
2. Describe the following characteristics of Java
Interpreted, Secure, High Performance.

Q.3 Attempt *ALL* from the following questions. **[12]**

(a) Answer the following [4]
(i) Two uses of super keyword.
(ii) Describe about final method and final arguments.

(b) Write IStack interface declaring push() and pop() methods, which can work on a stack of
integer numbers. Implement this interface using LinkedList as a storage of stack. Write code
for the following:
IStack interface [1]
LinkedList class [5]
IntegerStack class (implementation of IStack) [2]

OR

Q.3 Attempt *ALL* from the following questions. **[12]**

(a) We have a deck of playing cards (52 cards with 13 rank values and 4 suits) represented as a
Deck class. The playing cards have ranks: 2 to 10, Jack, Queen, King, and Ace; and suits as
Club, Diamond, Heart, and Spades. We would like to shuffle (arrange the cards in a random
order) an instance of the Deck class whenever we want. Write code for the following:
- *Deck* class with appropriate data members and the methods: shuffle() and [5]
displayCards().
- *DeckTest* that shows usage of the *Deck* class [1]

(b) Write code fragments for the following requirements:
(i) There is a Circle class containing the center (represented as x position and y position) and [3]
radius, all as integer numbers. Write the correct code to override the equals() method. We
should be able to invoke equals() method using a reference of *Object* class.
(ii) There are two classes: *Service* and *Consumer*. The *Consumer* class uses *Service* class. [3]
Answer the following in context of package
(a) The *Service* class is in a package *Weather*. If the *Consumer* class is also placed in the
same package, then what should be the first line in the definition of the *Consumer*
class?
(b) The *Service* class is in a package *Weather*. If the *Consumer* class is placed in a separate
package, then what statements should be written in the *Consumer* class and what
configuration of CLASSPATH should be done so that the *Consumer* class can access
the *Service* class?