Scheme of Evaluation IAT 1 – Sept. 2017

Sub:	ADVANCED JAVA					Sub Code:	15CS553	Branch:	CSE
Date:	21-09-2017	Duration:	90 min's	Max Marks:	50	Sem / Sec:	5th Sem / A, B, C		OBE

		MARKS	CO	RBT
1 (a)	Describe Enumerations. Explain values() and valueOf() methods in enumeration with program example.	[10]	CO1	L1

Eg: enum Season
{
Summer, Winter, Spring, Rainy
}

→ The identifiers Summer, Winter, Spring, Rainy are cal enumeration constants which are of enum type, i.e Season.

→ An instance of enum type cannot be created using ne

→ A variable of enum type can be created.

Eg: Season s;

⇒ Enumerations have 2 predefined methods:-

(a) values()

(b) valueOf()

→ values() : The values() method is used list al the enumeration constants of the same enumeration

general form :- public static enum-type[] values().

→ valueOf() : The valueOf() method is used to giv the enumeration constant specifically chosen (string as an output.

general form :- public static enum-type valueOf(

(or) valueOf(

Example Program

```
class ValuesOfDemo
{
    public static void main (String args[])
    {
        enum Season
        {
            Summer, Winter, Spring, Rainy
        }
        Season s;
        System.out.println("Here are all season constants: ");
        s = Season.values();
        System.out.println(s);
        System.out.println();
        System.out.println("ValueOf("Winter") = "+s.valueOf
                                                ("Winter"));
    }
}
```

Output :-

```
Here are all season constants
Summer
Winter
Spring
Rainy
ValueOf("Winter") = Winter.
```

2 (a)	List the Wrapper Classes in Java. Explain how Auto-boxing and Auto-unboxing occurs in expression with a program example.	[10]	CO1	L1

* TYPE WRAPPERS / WRAPPER CLASSES

The wrapper classes available in Java are :-

(i) Byte
(ii) Char
(iii) Short
(iv) Integer
(v) Long
(vi) Double
(vii) Float
(viii) Boolean

→ Wrapper classes are used to convert the primitiv to object and the object to primitive type.

→ Wrapper classes encapsulate the primitive typ inside the object.

* AUTOBOXING / UNBOXING IN EXPRESSIONS

→ Autoboxing :- Autoboxing is the process of encapsulati (boxed) the primitive types into its efficient type wrappers automatically.

→ Auto-unboxing :- Auto-unboxing is the process of extrac (unbox) the boxed values from the efficient type wrappers automatically.

→ Program

```
// Demonstration of Autoboxing/unboxing in expressions
class AutoBox
{
    public static void main (String args[])
    {
        Integer iObj1, iObj2;
        [illegible];
        iObj1 = 100;
        System.out.println ("Value of iObj1 = " +iObj1);
        ++iObj1;
        System.out.println ("Value of ++iObj1 = " +iObj1)
        iObj2 = iObj1 + (iObj1/3);
        System.out.println ("Value of iObj2 = " + iObj2);
        [illegible]
    }
}
```

3 (a) Describe how to override toString() method and obtain the dimensions of a Box passed through constructors and display it as a string when toString() method is called [10] CO3 L1

* OVERIDDING toString() METHOD

```
// Demonstration of overide toStrigng() and display result
   string.

Class Box
{
    int length;
    int breadth;
    int height;

    Box (int l , int b, int h)
    {
        length = l;
        breadth = b;
        height = h;
    }

    public String toString()
    {
        return " Length : " + length +
               " Breadth : " + breadth +
               " Height : " + height;
    }
}

    public static void main (String args[])
    {
        Box b = new Box (10 , 20 , 30);
        System.out.print (b);
    }
}
```

* Output :-

Length : 10 Breadth : 20 Height : 30

4 (a)	Explain the use of following methods of StringBuffer Class with a program example. [i] length() and capacity(). [ii] ensureCapacity(). [iii] setLength(). [iv] charAt and setCharAt().	[10]	CO3	L4

length() → specifies length of string

capacity() → ~~capac~~ specifies the place it ha

Sl.length() → default 16 characters.

Sl. capacity() → default 44 characters.

ensureCapacity() is used to set the values or capacity of StringBuffer.

Eg: ensureCapacity(100) → capacity is 100

setLength() is used to set the full length.

Eg: setLength(10) → only 10 char strin

5 (a)	List and explain different String Comparison methods available in Java and briefly explain each one of them with program examples	[10]	CO3	L4

String comparison methods are :-

(a) equals() and equalsIgnoreCase()

(b) RegionMatches()

(c) CompareTo()

(d) startsWith() and endsWith()

(e) equals() versus ==

(a) equals() and equalsIgnoreCase()

→ equals() checks for the equality comparison between 2 strings. It is case sensitive.

boolean equals()

→ equalsIgnoreCase() also checks for the equality comparison b/w 2 strings. Not case sensitive.
A to Z is same as a to z

boolean equalsIgnoreCase()

Eg:
```
String s1 = "hello";
String s2 = "hello";
String s3 = "GoodBye";
String s4 = "HELLO";

SOPln (s1 + "equals" + s2 + "→" + s1.equals(s2
SOPln (s1 + "equals" + s3 + "→" + s1.equals(s3
SOPln (s1 + "equals" + s4 + "→" + s1.equals(s4
SOPln (s1 + "equalsIgnoreCase" + s4 + "→"
                + s1.equalsIgnoreCa
```

Op:
```
s1 equals s2 → true
s1 equals s3 → false
s1 equals s4 → false
s1 equalsIgnoreCase → true
```

This method is used to compare a part of 1 string with a part of another string.

Eg:
```
String s1 = "Hello";
String s2 = "Hell";
SOPln (s1 + "equals" + s2 + "till 4 letters
        → " + s1.RegionMatches(0, s2, 3));
```

Op: s1 equals s2 till 4 letters → true.

(c) compareTo() : It is used mainly for sorting to check if strings are equal, less than or greater than.

Eg: Bubble sort.

(d) startsWith() : checks if a string starts with a particular substring

endsWith() : checks if a string ends a particular substring.

Eg: String S1 = "Football"

SOPln("S1 starts with Foot:" + S1.startsWith(

SOPln("S1 ends with ball:" + S1.endsWith("be

O/P: S1 starts with Foot : true

(e) equals() : compares the values / contens of 2 strings.

== :- compares the references of 2 string

Eg : String S1 = "Hello";

String S2 = new String ("Hello");

SOP("S1 equals S2:" + S1.equals(S2));

SOP("S1 == S2 : " + (S1 == S2));

O/P : S1 equals S2 : true

S1 == S2 : False.

6 (a) Briefly explain how to create annotations in Java. With the help of a program explain how annotations are obtained at Run Time by the use of Reflection. [10] CO1 L4

* ANNOTATIONS : enables to enforce supplemental informati into source file.

→ Annotations can be created using @interface and @Retention.

i.e.,

```
// Declaration of annotations
@interface
{
    String str();
    int val();
}
```

Program

```
import java.lang.annotations.*;
import java.lang.reflect.*;
@Retention (RetentionPolicy.RUNTIME)
@interface
{
    String str();
    int val();
}
```

```
public void myMeth()
{
    MyAnno (str = "Annotation Example", val = 100)
}
class Meta
{
    Meta ob = new Meta();
    Class c = ob.getClass();
    Method m = c.getMethod("myMeth");
    MyAnno anno = m.getAnnotations(MyAnno.class);
    System.out.println(anno.str() + " " + anno.val());
}
class MetaDemo
{
    public static void main(String arg[])
    {
        myMeth();
    }
}
```