KEMENTERIAN PENDIDIKAN TINGGI

POLITEKNIK MALAYSIA
METrO Tasek Gelugor

Politeknik METrO Tasek Gelugor
NO 25, LORONG KOMERSIAL 2,
PUSAT KOMERSIAL TASEK GELUGOR
13300 TASEK GELUGOR
PULAU PINANG

DFP30243
OBJECT ORIENTED PROGRAMMING

NAME:	**MUHAMMAD AFIQ MUHAIMIN BIN MOHD ZAINI**
CLASS:	**DDT3A**
REGISTRATION NO:	**32DDT20F2029**
LECTURER:	**PUAN HAZLEENA BINTI OSMAN** **PUAN RODZIAH BINTI IBRAHIM**
NAME OF TASK:	**LAB ACTIVITY 6**

LAB ACTIVITY 6:

String, String Buffer and Package

Learning Outcomes

This Lab sheet encompasses 12 activities (Activity 6A until 6L).

By the end of this lab, students should be able to:

- **Build Java programs using String and StringBuffer objects.**
- **Manipulate a String to a primitive data in Java programs.**
- **Construct program using method in String class.**
- **Create packages in Java Program**

Activity 6A

Activity Outcome: Demonstrates the String class constructors.

Procedure:
Step 1: Type the following program.

```
public class Act6A
{
      public static void main(String args[])
      {
            char charArray[] = {'b','i','r','t','h','d','a','y'};
            StringBuffer sb;
            String s1,s2,s3,s4, output;
            s1 = new String ("Hello");
            sb = new StringBuffer ("Welcome to Java Programming");

            //Use String constructors
            s2 = new String (s1);
            s3 = new String (charArray);
            s4 = new String (sb);

            //append Strings to output
            output = "s1 = " + s1 + "\ns2 = " + s2 + "\ns3 = " + s3 +
            "\ns4 = " + s4 + "\ns4 = " + s4;
            System.out.println(output);
      }
}
```

Step 2: Compile and execute the above program.

```java
//Demonstrate string class constructors
public class Act6A {
    Run | Debug
    public static void main(String args[]) {
        char charArray[] = { 'b', 'i', 'r', 't', 'h', 'd', 'a', 'y' };
        StringBuffer sb;
        String s1, s2, s3, s4, output;
        s1 = new String(original: "Hello");
        sb = new StringBuffer(str: "Welcome to Java Programming");
        // Use String constructors
        s2 = new String(s1);
        s3 = new String(charArray);
        s4 = new String(sb);
        // append Strings to output
        output = "s1 = " + s1 + "\ns2 = " + s2 + "\ns3 = " + s3 + "\ns4 = " + s4 + "\ns4 = " + s4;
        System.out.println(output);
    }
}
```

Step 3: Describe the output.

```
s1 = Hello
s2 = Hello
s3 = birthday
s4 = Welcome to Java Programming
s4 = Welcome to Java Programming
```

s1 shows Hello
s2 shows Hello too because of string constructor declaring s2 to take s1's string
s3 shows birthday from charArray[]
s4 shows StringBuffer data which is Welcome to Java Programming
Next s4 shows the original s4 output

Activity 6B

Activity Outcome: Demonstrates the length(), charAt() and getChars() method of the String class.

Procedure:
Step 1: Type the following program.

```
public class Act6B
{
	public static void main(String args[])
	{
		String s1;
		char chararray[];
		s1=new String("Hello there!");
		charArray= new char [5];
		System.out.println ("s1 : " + s1);
		System.out.println ("Length of s1 : " + s1.length());
		System.out.println ("Character at 3rd position :
		"+s1.charAt(2));
		s1.getChars(0,5,charArray,0);
		System.out.print("The character array is : ");
		for(int i=0;i<charAray.length;i++)
			System.out.print(" " + charArray[i]);
		System.out.println();
	}
}
```

Step 2: Compile and execute the above program.

```
//Demonstrates the length(), charAt() and getChars() method of the String class.
public class Act6B {
    Run | Debug
    public static void main(String args[]) {
        String s1;
        char charArray[];
        s1 = new String(original: "Hello there!");
        charArray = new char[5];
        System.out.println("s1 : " + s1);
        System.out.println("Length of s1 : " + s1.length());
        System.out.println("Character at 3rd position : " + s1.charAt(index: 2));
        s1.getChars(srcBegin: 0, srcEnd: 5, charArray, dstBegin: 0);
        System.out.print(s: "The character array is : ");
        for (int i = 0; i < charArray.length; i++) {
            System.out.print(" " + charArray[i]);
            System.out.println();
        }
    }
}
```

Step 3: Describe the output.

```
s1 : Hello there!
Length of s1 : 12
Character at 3rd position : l
The character array is :  H
 e
 l
 l
 o
```

S1 is outputting Hello there!
Length of s1 is displayed using s1.length()
Character at 3rd position is displayed using s1.charAt() method and specifically on index 2
Character array is shown using for loop comparing to length of charArray[] and display each of the word inside charArray one by one

Activity 6C

Activity Outcome: Demonstrates the methods equals(), compareTo() of the String class

Procedure:
Step 1: Type the following program.

```
public class Act6C
{
      public static void main(String args[])
      {
            String s1,s2,s3;
            s1=new String("Hello");
            s2=new String("Good Bye");
            s3=new String("Happy Birthday");
            System.out.println("s1 = " + s1+"\ns2 = " + s2+ "s3 = " + s3);

            //Test for equality
            if (s1.equals("Hello")
                  System.out.println("s1 equals \"Hello\" ");
            else
                  System.out.println("s1 does not equal \"Hello\" ");

            //Test for equality with == operator
            if (s1=="Hello")
                  System.out.println("s1 equals \"Hello\" ");
            else
                  System.out.println("s1 does not equal \"Hello\" ");

            //Test compareTo
            System.out.println("s1 compare to \"Hello\" is
            "+s2.compareTo("Hello"));
            System.out.println("s2 compare to s3 is " + s2.compareTo(s3));
      }
}
```

Step 2: Compile and execute the above program.

```java
//Demonstrates the methods equals(), compareTo() of the String class
public class Act6C {
    Run | Debug
    public static void main(String args[]) {
        String s1, s2, s3;
        s1 = new String( original: "Hello");
        s2 = new String( original: "Good Bye");
        s3 = new String( original: "Happy Birthday");
        System.out.println("s1 = " + s1 + "\ns2 = " + s2 + "s3 = " + s3);

        // Test for equality
        if (s1.equals( anObject: "Hello")) {
            System.out.println( x: "s1 equals \"Hello\" ");
        } else {
            System.out.println( x: "s1 does not equal \"Hello\" ");
        }

        // Test for equality with == operator
        if (s1 == "Hello") {
            System.out.println( x: "s1 equals \"Hello\" ");
        } else {
            System.out.println( x: "s1 does not equal \"Hello\" ");
        }
        // Test compareTo
        System.out.println("s1 compare to \"Hello\" is " + s2.compareTo( anotherString: "Hello"));
        System.out.println("s2 compare to s3 is " + s2.compareTo(s3));
    }
}
```

Step 3: Describe the output.

```
s1 = Hello
s2 = Good Byes3 = Happy Birthday
s1 equals "Hello"
s1 does not equal "Hello"
s1 compare to "Hello" is -1
s2 compare to s3 is -1
```

S1 is displaying Hello
S2 is displaying Good Bye
S3 is displaying Happy birthday
First If statement check for equality in s1 to Hello
If it's the same then output S1 equals hello
If not output s1 does not equals hello

Next code check for equality but with == operator
If it's the same then output S1 equals hello
If not output s1 does not equals hello

Lastly code is comparing s1 to s2 and s2 compared to s3

Activity 6D

Activity Outcome: Demonstrates the String class index methods

Procedure:

Step 1: Type the following program.

```
public class Act6D
{
   public static void main(String args[])
   {
      String letters=”This is a test”;
      int value1, value2, value3, value4;
      value1=letters.indexOf(‘s’);
      System.out.println(“1ST occurrence of \”s\” in a string “+“ at index :“
            +value1);
      value2=letters.indexOf(”is”);
      System.out.println(“1ST occurrence of \”is\” in a string “+“ at index :“
            +value2);
      value3=letters.lastIndexOf(‘s’);
      System.out.println(“last occurrence of \”s\” in a string “+“is at index:“
            +value3);
      value4=letters.lastIndexOf(”is”);
      System.out.println(“last occurrence of \”s\” in a string “+“is at index:“
            +value4);
   }
}
```

Step 2: Compile and execute the above program.

```
//Demonstrates the String class index methods
public class Act6D {
    Run | Debug
    public static void main(String args[]) {
        String letters = "This is a test";
        int value1, value2, value3, value4;
        value1 = letters.indexOf( ch: 's');
        System.out.println("1ST occurrence of \"s\" in a string " + " at index :" + value1);
        value2 = letters.indexOf( str: "is");
        System.out.println("1ST occurrence of \"is\" in a string " + " at index :" + value2);
        value3 = letters.lastIndexOf( ch: 's');
        System.out.println("last occurrence of \"s\" in a string " + "is at index:" + value3);
        value4 = letters.lastIndexOf( str: "is");
        System.out.println("last occurrence of \"s\" in a string " + "is at index:" + value4);
    }
}
```

Step 3: Describe the output.

```
1ST occurrence of "s" in a string  at index :3
1ST occurrence of "is" in a string  at index :2
last occurrence of "s" in a string is at index:12
last occurrence of "s" in a string is at index:5
```

First line checks when “s” is first found in the defined string. The Answer is index 3
Second line checks when is “is” first found in the defined string. The Answer is index 2
Third line checks when “s” is last found in the defined string. The Answer is index 12
Fourth line checks when “s” is last found in the defined string. The Answer is index 5

Activity 6E

Activity Outcome: Demonstrates the String class substring methods.

Procedure:

Step 1: Type the following program.

```
public class Act6E
{
      public static void main(String args[])
      {
        String s1=”I do shopping in Sunway Carnival Mall”;
        String s2=s1.substring(2,10);
        String s3=s1.substring(17);
        System.out.println(“substring from index 2 to 10 is: “+s2);
        System.out.println(“substring from index 17 to end is: “+ s3);
      }
}
```

Step 2: Compile and execute the above program.

```
//Demonstrates the String class substring methods
public class Act6E {
    Run | Debug
    public static void main(String args[]) {
        String s1 = "I do shopping in Sunway Carnival Mall";
        String s2 = s1.substring( beginIndex: 2,  endIndex: 10);
        String s3 = s1.substring( beginIndex: 17);
        System.out.println("substring from index 2 to 10 is: " + s2);
        System.out.println("substring from index 17 to end is: " + s3);
    }
}
```

Step 3: Differentiate the substring method operation.

```
substring from index 2 to 10 is: do shopp
substring from index 17 to end is: Sunway Carnival Mall
```

First substring on **s2** is extracting the string in s1 starting from **index 2** and ending on **index 10**

Second substring on **s3** extracting the string in s1 starting from **index 17** and ending when the index ends

Activity 6F

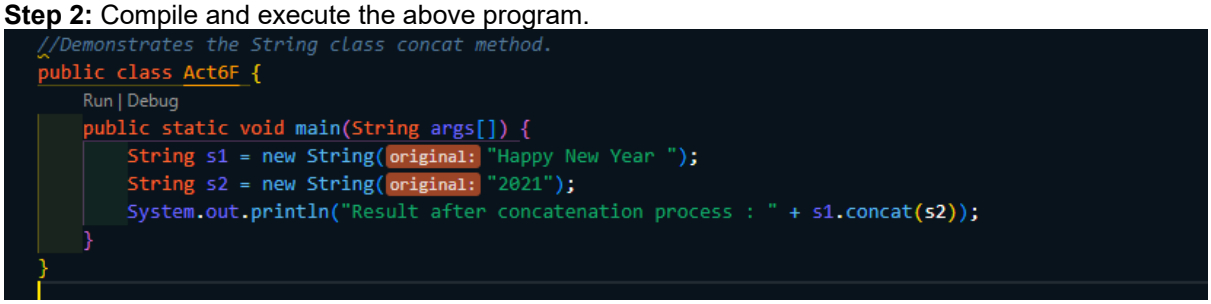
Activity Outcome: Demonstrates the String class concat method.

Procedure:

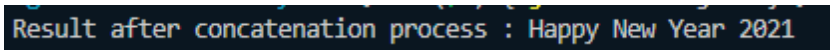
Step 1: Type the following program.

```
public class Act6F
{
	public static void main(String args[])
	{
		String s1= new String("Happy New Year ");
		String s2=new String("2021");
		System.out.println("Result after concatenation process : " +
		s1.concat(s2));
	}
}
```

Step 2: Compile and execute the above program.



Step 3: State the output.



Activity 6G

Activity Outcome: Demonstrates StringBuffer constructors.

Procedure:
Step 1: Type the following program.

```
public class Act6G
{
	public static void main(String args[])
	{
		StringBuffer bf1, bf2,bf3;
		bf1= new StringBuffer();
		bf2= new StringBuffer(10);
		bf3= new StringBuffer("PoliteknikUngku Omar");
		System.out.println("Buffer 1 : \"" + bf1.toString()+ " \"");
		System.out.println("Buffer 1 : \"" + bf2.toString()+ " \"");
		System.out.println("Buffer 1 : \"" + bf3.toString()+ " \"");
	}
}
```

Step 2: Compile and execute the above program.

```
//Demonstrates StringBuffer constructors.
public class Act6G {
    Run | Debug
    public static void main(String args[]) {
        StringBuffer bf1, bf2, bf3;
        bf1 = new StringBuffer();
        bf2 = new StringBuffer(capacity: 10);
        bf3 = new StringBuffer(str: "PoliteknikUngku Omar");
        System.out.println("Buffer 1 : \"" + bf1.toString() + " \"");
        System.out.println("Buffer 1 : \"" + bf2.toString() + " \"");
        System.out.println("Buffer 1 : \"" + bf3.toString() + " \"");
    }
}
```

Step 3: Differentiate the various type of StringBuffer constructors.

```
Buffer 1 : " "
Buffer 1 : " "
Buffer 1 : "PoliteknikUngku Omar "
```

Constructor	Description
StringBuffer()	It creates an empty String buffer with the initial capacity of 16.
StringBuffer(String str)	It creates a String buffer with the specified string.
StringBuffer(int capacity)	It creates an empty String buffer with the specified capacity as length.

Activity 6H

Activity Outcome: Demonstrates the length() and capacity() methods of the StringBuffer class.

Procedure:

Step 1: Type the following program.

```
public class Act6H
{
	public static void main(String args[])
	{
		StringBuffer bf = new StringBufffer(“Hello, how are you?”);

		System.out.println(“Buffer ” + bf.toString() + “\nlength = ”
		+bf.length()+ “\ncapacity = ”+bf.capacity());

		bf.ensureCapacity(75);

		System.out.println(“\n\nNew Capacity = ” + bf.capacity());

		bf.setLength(10);
		System.out.println(“New length = ” + bf.length() +
		“\n Buffer : ” +bf.toString());
	}
}
```

Step 2: Compile and execute the above program.

Program

```
//Demonstrates the length() and capacity() methods of the StringBuffer class.
public class Act6H {
    Run | Debug
    public static void main(String args[]) {
        StringBuffer bf = new StringBuffer( str: "Hello, how are you?");
        System.out.println("Buffer " + bf.toString() + "\nlength = " + bf.length() + "\ncapacity = " + bf.capacity());
        bf.ensureCapacity( minimumCapacity: 75);
        System.out.println("\n\nNew Capacity = " + bf.capacity());
        bf.setLength( newLength: 10);
        System.out.println("New length = " + bf.length() + "\n Buffer : " + bf.toString());
    }
}
```

Output

```
Buffer Hello, how are you?
length = 19
capacity = 35


New Capacity = 75
New length = 10
 Buffer : Hello, how
```

Activity 6I

Activity Outcome: Convert a String to a primitive data in Java programs.

Procedure:
Step 1: Type the following program.

```
import java.io.*;
class Act6I
{
	public static void main (String args[]) throws IOException
	{
		BufferedReader stdin = new BufferedReader(new
		InputStreamReader(System.in));
		String str;
		int num;		// declares an int variable

		System.out.println("Enter an integer:");
		str = stdin.readLine();
		num = Integer.parseInt(str);
		System.out.println("Integer Value: "+num);
	}
}
```

Step 2: Compile and execute the above program.

```
//Convert a String to a primitive data in Java programs.
import java.io.*;

class Act6I {
    Run | Debug
    public static void main(String args[]) throws IOException {
        BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));
        String str;
        int num; // declares an int variable

        System.out.println(x: "Enter an integer:");
        str = stdin.readLine();
        num = Integer.parseInt(str);
        System.out.println("Integer Value: " + num);
    }
}
```

Step 3: State the output

```
Enter an integer:
50
Integer Value: 50
```

PACKAGE

Activity 6J (i)

Activity Outcome: Demonstrate how to create a package in Java program

Procedure:
Step 1: Declare the package in the first line of the source file. Package name must be in lowercase letter.
Step 2: Declare the class as *public*, so that it can be used in other packages.
Step 3: Define the methods of the class.
Step 4: Create a subdirectory under the current directory.
Step 5: Name the subdirectory same as the package name.
Step 6: Store the source file in the subdirectory.
Step 7: Compile the file. This file creates .class file in the subdirectory.

Activity 6J (ii)

Activity Outcome: Declare the package and definition of the class in the package.

Procedure :
Step 1: Create a *subdirectory* pack1 in the current directory.
Step 2: Save the program as *ClassDdt*.j*ava* in the *pack1* subdirectory. For example if *d:*\ is the current directory, then create *d:**pack1*and save *ClassDdt.java* in it.

```
package pack1;
public class ClassDdt
{
        public void show ()
        {
             int x,y,result;
             x=10;
             y=5;
             result=x*y;
             System.out.println ("The result is "+result);
        }
}
```

Step 3: Compile the program and save it the *pack1* subdirectory.

Activity 6J (iii)

Activity Outcome: Demonstrate how to import class from the created package

Procedure:

Step 1: Save the program as packtest.java in the current directory.
Step 2: If d: is the directory in which the packtest.java is and the user-defined package is in d:\pack1, then set the classpath as set CLASSPATH=%CLASSPATH%;d:\pack1; in the current directory. Type the above command in the command prompt and execute it to set the path.

```
import pack1. ClassDdt;
class Packtest1
{
        public static void main(String args[])
        {
             ClassDdt c1 = new ClassDdt ();
             c1.show();
        }
}
```

Step 3: Compile and run the program.

Code

ClassDdt.java

```
package pack1;

public class ClassDdt {
    public void show() {
        int x, y, result;
        x = 10;
        y = 5;
        result = x * y;
        System.out.println("The result is " + result);
    }
}
```

pack1.java

```
public class pack1 {
}
```

Packtest1.java

```
import pack1.ClassDdt;

class Packtest1 {
    Run | Debug
    public static void main(String args[]) {
        ClassDdt c1 = new ClassDdt();
        c1.show();
    }
}
```

Output