Sudhamshu B N
1DS18EC091
6B

Introduction to JAVA

1.)

a) c) package

b) d) All of the mentioned

c) c) import pkg.*

d) ~~d)~~ A package can be renamed without renaming the directory in which the classes are stored

e) b) java

f) b) append()

g) a) length()

h) a) IO Exception

i) d) StringBuffer(int size, String str)

j) a) reverse()

2) Packages:

Packages in java core way to encapsulate a group of classes, interfaces, enumerations, annotations, and sub packages.

Using packages gives advantages like.,

→ Reusability
→ Name conflicts
→ Controlled Access
→ Data Encapsulation
→ Maintainance

* There are
  i) Built in Packages
  ii) User defined Packages

Usage/Creating a Package in JAVA

* For example we can create a package easily like,

```
package MyPackage;
```

To create a class inside package we declare name as first statment of program.

* User Defined Package

Ex:-

```
package MyPackage

public class Compare{
    int n1, n2;
    Compare (int n, int m){
        n1 = n;
        n2 = m;
    }
public void getmax(){
    if (n1>n2){
        System.out.println("maximum value "+n1);
```

```
else{
    System.out.println("Maximum value" + n2);
  }
}

public static void main(String args[]){
        Compare current[] = new Compare[3];
        current[1] = new Compare(5, 10);
        current[2] = new Compare(123, 120);
for(int i=1; i<3; i++){
        current[i].getmax();
    }}}
```

output:

Maximum value of two numbers is 10

Maximum value of two numbers is 123

* If it is a built in package, then.

Ex: package DSCE;

```
import java.util.ArrayList;
class BuiltinPackage{
    public static void main(String[] args){
        ArrayList<Integer> mylist = new ArrayList<>(3);
    mylist.add(3);
    mylist.add(2);
    mylist.add(1);
    System.out.println("The elements are:"+mylist);
    }}
```

Output:

The elements are: [3,2,1]

30) Special String Operators

* String Literals
* String Concatenation
* String Concatenation with other Data Types
* String Conversion and toString().

String Literals : Each time you create a string literal the JVM checks the string constant pool first. If the string already exists in the pool a reference to the pooled instance is returned. If string doesn't exist, then new String instance is created & placed in the pool

[Ex: String s = "welcome";
String t = "welcome"; // will not be created.]

- In 1st statement new object is created.
- In 2nd statement "welcome" is already there in pool so it is not created.

String Concatenation.

In Java + operator is used for string concatenation

for example,

[Ex: String age = "9";
String s = "He is " + age + " years old";
System.out.println(s);]

Output : He is 9 years old.

For very long strings also string concatenation works. but with java.lang.String.concat(String str) method.

String Concatenation with other Data types

Other data like int can be concatenated with string

[Ex: int age = 9;
String s = "He is " + age + "years old" ;
System.out.println(s);]

The output is same as before. This is because the int value age is automatically converted into its string representation within a string object. + operator is used even here.

But while doing operations like

String s = "four:" + 2+2 ⇒ output four:22 X

String s = "four:" + (2+2) ⇒ output four:4 V

String conversion and toString()

Value of & toString() methods give the value of given string.

[Ex:-
public static String valueOf(boolean b);
public static String valueOf(char c)] } valueOf()

toString() is used to convert other data types to string.

} toString()

[Ex: general method form

String toString(_);

then println() can be used to print the string that is converted by using toString() method.

4) Default imported package: java.lang package

Package import is a feature in java which allows to reuse the classes available in a package. Java provides import keyword to import another package.

Syntax for importing package or class

→ // import all classes in a package
import packagename.*;

→ // import specific class
import packagename.classname

→ // import all classes of subpackage.
import packagename.subpackage name

→ // import specific class of a subpackage class
import packagename.subpackagename.classname.

Package import statement must come after package name declaration & before the start of your class. Otherwise import statement can be first statement.

~~Accessing classes~~

Example: Built in Package

Syntax: import package.name.class;
import package.name.*;

Example:
```
import java.util.Scanner;
class MyClass {
    public static void main(String[] args) {
        Scanner myObj = new Scanner(System.in);
        System.out.println("Enter user name");
    }
}
```

Example : User defined Packages

└ root
  └ mypack
    └ MyPackageClass.java.

```
package mypack;
 class MyPackageClass {
      public static void main(String [] args) {
       System.out.println("This is my package");
      }
```

In another program this package can be imported like,

```
package Java;
import mypack.*; // ~~Eg~~ importing user defined package
class mypackageclass {
      public static void main (String args [ ]) {
        MyPackageClass obj = new MyPackageClass ();
        obj.main ();
      }
}
```

Output : This is My package.

Even user defined packages can be imported using import keyword.

7) Find the length of string using length() method.

```
import java.util.Scanner;
public class JavaProgram
  {
    public static void main (String args[])
      { String str;
        int len;
        Scanner scan = new Scanner (System.in);
        System.out.print ("Enter Your Name:");
        str = scan.nextLine();
        len = str.length();
        System.out.println ("Length of the entered String is " + len);
      }
  }
```

Enter Your Name : Sudhanshu B N
Length of the entered string is 13

Without using length() method

```
class Length of String main {
    public static void main (String args[]){
      String helloworld = "Hello world!";
      System.out.println ("length :" + getLength(helloworld));
    }
```

```
public static int getLengthOfStringWithCharArray(String str)
    { int length=0;
     char[] strCharArray = str.toCharArray();
     for(char c: strCharArray)
      { length++;
     }
      return length;
     }
```

length: 12