SLIIT UNI
THE KNOWLEDGE UNIVERSITY

Sri Lanka Institute of Information Technology

Faculty of Computing

IT1120 - Introduction to Programming

Dr.Junius Anjana

Year 01 and Semester 01

Lecture 3

Java Constants, Char Data Type, User Input Handling

Java Constants

- In Java, a constant refers to a variable that holds a fixed value throughout the program's execution.
- Using final and static keyword, we can declare a constant.
- Here're few examples of creating constants

```
public static final double PI = 3.14159; //for the mathematical value pi
public static final int MAX_TRIES = 3; // Maximum number of attempts allowed
public static final String DEFAULT_NAME= "sliit";
```

Char Data Type

- Characters are normally stored in variable type char
- Characters can be stored in any integer type variable too
- Characters can be treated as either an *integer* or a *character*
- Many computers today use ASCII (American Standard Code for Information Interchange) character set

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

        char ch = 'A';
        int asciiValue = ch;

        System.out.println("Character " + ch + " has the value " +asciiValue);
    }
}
```

Output

```
Character A has the value 65
```

Relational Operator (== , !=) with Decimal values

- When a decimal number is storing as a float or double data type value, sometimes the computer round off the excess digits. This is called truncation.
- Due to the truncation error inherent in storing float and double values, these values should never be compared using == or != relational operators.

Java Libraries

- Java libraries are collections of pre-written code that add functionality to your java programs.
- Think of them as tools that you can use to save time and effort when building your Java programs.
- There are two main types of libraries:
  1. Standard libraries
  2. External libraries

1. **Standard libraries / Java class libraries**
   These come with Java by default. They provide a bunch of useful classes and methods.
   Like:
   - Reading user input
   - Working with files
   - Making calculations

2. **External libraries**
   These are like extra libraries that provide specialized functionality for specific tasks.
   - Graphics for games
   - Connecting to databases
   - Working with the internet
   - User-written libraries

Using libraries in Java program

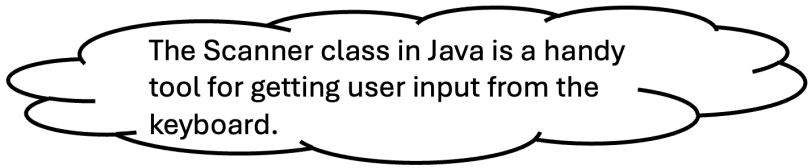
- By including **import** statements in your Java code, we can use standard libraries.
- **java.util** is a standard library in Java.
- Let's see, how **java.util** is used in your Java code.

Example : `import java.util.*;`

This **import** statement specifies that import all class files from the **util** standard library.

Example : `import java.util.Scanner;`

This **import** statement specifies that import only **Scanner** class file from the **util** standard library.



How to take user inputs from the keyboard

- In Java, the Scanner class is used to capture user input from the keyboard.

Let's see how it can be done...

1. First, you need to tell your program that you want to use Scanner class in java.util standard library. You do this with an import statement:

```
import java.util.Scanner;
```

2. Next, you create a Scanner object.

```
Scanner input = new Scanner(System.in);
```

Here,

- Name of the object is **input**. This can be any name.
- Use the **new** keyword to create a new Scanner object.
- Pass **System.in** as an argument, which tells the scanner to read the user input from the keyboard.

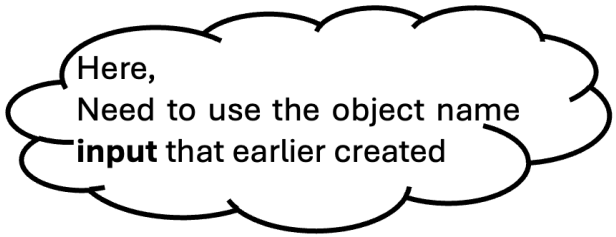
3. Now it's time to capture the user input. Here comes the magic! You need to use appropriate function from the Scanner class to capture the user input and store it in a variable. It depends on the data type of the user input.

For Integers: to read an integer number from the keyboard

```
int number = input.nextInt();
```

For double:

```
double length = input.nextDouble();
```



For Integers:

```
float price = input.nextFloat();
```

For a single character: It is slightly different. It reads the first character from the user.

```
char garde = input.next().charAt(0);
```

For Strings:

```
String name = input.next();
String name = input.nextLine();
```

For Strings:

```
String name = input.next();
String name = input.nextLine();
```

Exercises

Enter the radius of a circle from the keyboard and display the circumference and the area.

1. Write the relevant pseudocode.
2. Convert it to Java program.

Pseudocode

```
MAIN:

        DEFINE radius, area, circumference AS FLOAT

        INPUT radius

        circumference = 2 * 22.0 / 7 * radius

        area = 22.0 / 7 * radius * radius

        PRINT “Area is ” , area

         PRINT “Circumference is ” , circumference

ENDMAIN
```

Java Code

```
import java.util.Scanner;

public class lecture3Exercise
{
        public static void main(String[] args)
        {
                //Declare variables
                double radius, area, circumference;

                //Create an object called input to Scanner class
                Scanner input = new Scanner(System.in);

                //Take radius value as a user input
                System.out.println("Enter radius : ");
                radius = input.nextDouble();

                //Calculations
                circumference = 2 * 22.0 / 7 * radius;
                area = 22.0 / 7 * radius * radius;

            //Display output
            System.out.println("Area is " + area);
            System.out.println("Circumference is " + circumference);
        }
```

Thank You!