Welcome to My Presentation

My Presentation Topic is

Keyword and Data type

Presented By:

Name : Tushar Sarkar

Student ID: 18CSE035

Second Year First Semester

Department Of CSE,BSMRSTU,

Gopalganj-8100.

What is java keyword

- A Java keyword is one of 50 reserved terms that have a special function and a set definition in the Java programming language.

- The fact that the terms are reserved means that they cannot be used as identifiers for any other program elements, including classes, subclasses, variables, methods and objects.

List of java keyword

boolean	**double**
break	else
case	extends
char	final
continue	float
default	for
class	if
do	import

Java Boolean Keyword

- In Java, the boolean keyword is a primitive data type.
- It is used to store only two possible values, either true or false.
- It specifies 1-bit of information and its "size" can't be defined precisely.

```
public class BooleanExample{
        public static void main(String args[]){
                boolean b1=true;
                boolean b2=false;
                boolean b3=(b1==b2);
                System.out.println(b1);
                System.out.println(b2);
                System.out.println(b3);
        }
}
```

Java Break Keyword

- When a break statement is encountered inside a loop, the loop is immediately terminated and the program control resumes at the next statement following the loop.
- The Java *break* statement is used to break loop or switch statement

```
public class BreakExample{
        public static void main(String args[]){
                // using for loop
                for(int i=1; i<=10; i++){
                        If(i==5){
                                // breaking the loop
                                break;
                        }
                        System.out.println(i);
                }
        }
}
```

Java case Keyword

- The Java case keyword is a conditional label which is used with the switch statement.
- It contains a block of code which is executed only when the switch value matches with the case.
- A switch statement can contain multiple case labels.
- Each case label must hold a different value.
- The case label can contain the break statement that terminates the flow of the execution;

```
switch(expression){
        case value 1:
            //code to be excuted;
            break; // optional
        case value 2 :
            //code to be excuted;
            break; // optional
        defulat:
            // code to be excuted if all acses are not matched;
}
```

Java Continue Keyword

- The continue statement is used in loop control structure when you need to jump to the next iteration of the loop immediately.
- It can be used with for loop or while loop.
- The Java *continue statement* is used to continue the loop.
- It continues the current flow of the program and skips the remaining code at the specified condition

```
public class continueExample{
        public static void main(String args[]){
                // for loop
                for(int i=1; i<=10; i++){
                        If(i==5){
                                // using continue statement
                                continue;
                        }
                        System.out.println(i);
                }
        }
}
```

Java If-else Keyword

- The Java if-else statement also tests the condition.
- It executes the *if block* if condition is true otherwise *else block* is executed.

```
public class IfElseExample{
        public static void main(String args[]){
                //define a variable
                int number=13;
                //check if the number is devisible by 2 or not
                if(number%2==0){
                        System.out.println("Eveb Number");
                }
                else{
                        System.out.prinrtln("Odd Number");
                }
        }
}
```

Java data type

- Java has two categories of data

1. Primitive Data Type: such as boolean, char, int, short, byte, long, float, and double.
2. Non-Primitive Data Type or Object Data type: such as String, Array, etc.

Primitive data types

1. Boolean : Boolean data type represents only one bit of information **either true or false**, but the size of the boolean data type is **virtual machine-dependent**. Values of type boolean are not converted implicitly or explicitly (with casts) to any other type. But the programmer can easily write conversion code.

- **Syntax:** boolean booleanVar;
- **Size:** virtual machine dependent
- **Values**: true, false
- **Default Value:** false

Primitive data types

2. **Byte :**The byte data type is an 8-bit signed two's complement integer. The byte data type is useful for saving memory in large arrays.

- **Syntax:** byte byteVar;
- **Size:** 1 byte (8 bits)
- **Values**: -128 to 127
- **Default Value:** 0

3. **Short :** The short data type is a 16-bit signed two's complement integer. Similar to byte, use a short to save memory in large arrays, in situations where the memory savings actually matters.

- **Syntax: short shortVar;**
- **Size:** 2 byte (16 bits)
- **Values**:-32, 768 to 32, 767 (inclusive)
- **Default Value:** 0

Primitive data types

4. int : It is a 32-bit signed two's complement integer.

- **Syntax:** int intVar;
- **Size:** 4 byte (32 bits)
- **Values**:--2, 147, 483, 648 to 2, 147, 483, 647 (inclusive)
- **Default Value:** 0

5. long : The long data type is a 64-bit two's complement integer.

- **Syntax:** long longVar;
- **Size:** 8 byte (64 bits)
- **Values**:-9, 223, 372, 036, 854, 775, 808
  to 9, 223, 372, 036, 854, 775, 807
- **Default Value:** 0

Primitive data types

6. float : The float data type is a single-precision 32-bit IEEE 754 floating-point. Use a float (instead of double) if you need to save memory in large arrays of floating-point numbers.

- **Syntax:** float floatVar;
- **Size:** 4 byte (32 bits)
- **Values**: upto 7 decimal digits
- **Default Value:** 0.0

5. **double**: The double data type is a double-precision 64-bit IEEE 754 floating-point. For decimal values, this data type is generally the default choice.

- **Syntax:** double doubleVar;
- **Size:** 8 byte (64 bits)
- **Values**:upto 16 decimal digits
- **Default Value:** 0.0

Primitive data types

8. char : The char data type is a single 16-bit Unicode character.

- **Syntax:** char charVar;
- **Size:** 2 byte (16 bits)
- **Values**: '\u0000' (0) to '\uffff' (65535)
- **Default Value:** '\u0000'

Non-Primitive Data Type or Reference Data Types

The **Reference Data Types** will contain a memory address of variable value because the reference types won't store the variable value directly in memory. They are **strings**, **objects**, arrays, etc.

- String: Strings are defined as an array of characters. The difference between a character array and a string in Java is, the string is designed to hold a sequence of characters in a single variable whereas, a character array is a collection of separate char type entities.
- Unlike C/C++, Java strings are not terminated with a null character.
  Below is the basic syntax for declaring a string in Java programming language.

Syntax: <String_Type> <string_variable> = "<sequence_of_string>";

Non-Primitive Data Type or Reference Data Types

The **Reference Data Types** will contain a memory address of variable value because the reference types won't store the variable value directly in memory. They are **strings**, **objects**, arrays, etc.

- **Array:** An array is a group of like-typed variables that are referred to by a common name. Arrays in Java work differently than they do in C/C++. The following are some important points about Java arrays.
- In Java, all arrays are dynamically allocated. (discussed below)
- Since arrays are objects in Java, we can find their length using member length. This is different from C/C++ where we find length using size.
- A Java array variable can also be declared like other variables with [] after the data type.
- The variables in the array are ordered and each has an index beginning from 0.
- Java array can be also be used as a static field, a local variable or a method parameter.
- The **size** of an array must be specified by an int value and not long or short

Thank you