Course title	: **CSE2005**
Course title	: **Object Oriented Programming**
Module	: **0**
Topic	: **3**

Java – Data Types

Objectives

This session will give the knowledge about

- Key points about Java
- Data types in Java
- Variables in Java

The Java API

- An application programming interface(API), in the framework of java, is a collection of prewritten packages, classes, and interfaces with their respective methods, fields and constructors

- The Java API, included with the JDK, describes the function of each of its components

- In Java programming, many of these components are pre-created and commonly used

The Java Buzzwords

- Simple
- Object-Oriented
  - Supports encapsulation, inheritance, abstraction, and polymorphism
- Distributed
  - Libraries for network programming
  - Remote Method Invocation
- Architecture neutral
  - Java Bytecodes are interpreted by the JVM

The Java Buzzwords

- Secure
  - Difficult to break Java security mechanisms
  - Java Bytecode verification
  - Signed Applets

- Portable
  - Primitive data type sizes and their arithmetic behavior specified by the language
  - Libraries define portable interfaces

- Multithreaded
  - Threads are easy to create and use

Java Keywords

abstract	continue	for	new	switch
assert	default	goto	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while

Primitive Data Types

Data Type	Size (in bits)	Minimum Range	Maximum Range	Default Value (for fields)
byte	8	-128	+127	0
short	16	-32768	+32767	0
int	32	-2147483648	+2147483647	0
long	64	-9223372036854775808	+9223372036854775807	0L
float	32	1.40E-45	3.40282346638528860e+38	0.0f
double	64	4.94065645841246544e-324d	1.79769313486231570e+308d	0.0d
char	16		0 to 65,535	'\u0000'
boolean	1	NA	NA	false

Types of Variables

The Java programming language defines the following kinds of Variables:

- Local Variables
  - Tied to a method
  - Scope of a local variable is within the method
- Instance Variables (Non-static)
  - Tied to an object
  - Scope of an instance variable is the whole class
- Static Variables
  - Tied to a class
  - Shared by all instances of a class

Quiz

1. What will be the result, if we try to compile and execute the following code?

```
class Test
{
    public static void main(String [ ] ar)
    {
        int for=2;
        System.out.println(for);
    }
}
```

Quiz

2. What will be the result, if we try to compile and execute the following code?

```
class Test
{
    public static void main(String [ ] ar)
    {
        byte b=128;
        System.out.println(b);
    }
}
```

Quiz

3. What will be the result, if we try to compile and execute the following code?

```
class Test
{
    public static void main(String ar[])
    {
        float f=1.2;
        boolean b=1;
        System.out.println(f);
        System.out.println(b);
    }
}
```

Quiz

4. What will be the result, if we try to compile and execute the following code?

```
class Test
{
    public static void main(String ar[])
    {
        double d=1.2D;
        System.out.println(d);
    }
}
```

Quiz

5. What will be the result, if we try to compile and execute the following code?

```
class Test
{
    public static void main(String [ ] ar)
    {
        int a=10,b=017,c=0X3A;
        System.out.println(a+","+b+","+c);
    }
}
```

Quiz

6. What will be the result, if we try to compile and execute the following code?

```
class Test
{
    public static void main(String [ ] args)
    {
        int 9A=10;
        System.out.println(9A);
    }
}
```

Quiz

7. What will be the result, if we try to compile and execute the following code?

```
class Test
{
    public static void main(String [ ] args)
    {
        int x;
        System.out.println(x);
    }
}
```

Quiz

8. Match the following table:

DATA TYPES	SIZE(bytes)
char	4
byte	2
int	1
double	8

Course code : **CSE1004**

Course title : **Problem Solving using Java**

Java – Type Casting

Objectives

This session will give the knowledge about to

- Work with type casting in Java

Introduction to Type casting

- Type casting is when you assign a value of one primitive data type to another type.

- In Java, there are two types of casting:

- Widening Casting (automatically) - converting a smaller type to a larger type size

  byte -> short -> char -> int -> long -> float -> double

- Narrowing Casting (manually) - converting a larger type to a smaller size type

  double -> float -> long -> int -> char -> short -> byte

Widening Casting

```
public class MyClass {
  public static void main(String[] args) {
    int myInt = 9;
    double myDouble = myInt; // Automatic casting: int to double

    System.out.println(myInt);      // Outputs 9
    System.out.println(myDouble);   // Outputs 9.0
  }
}
```

Narrowing or Explicit Conversion

If we want to assign a value of larger data type to a smaller data type we perform explicit type casting or narrowing.

- This is useful for incompatible data types where automatic conversion cannot be done.
- Target data type have to be represented in ( ) next to the = sybmol

Explicit Conversion Example

```
public class MyClass {
  public static void main(String[] args) {
    double myDouble = 9.78;
    int myInt = (int) myDouble; // Manual casting: double to int

    System.out.println(myDouble);   // Outputs 9.78
    System.out.println(myInt);      // Outputs 9
  }
}
```

Explicit Conversion Example

```
class Test
{
    public static void main(String[] args)
    {
        double d = 100.04;

        //explicit type casting
        long l = (long)d;

        //explicit type casting
        int i = (int)l;
```

Explicit Conversion Example

```
        System.out.println("Double value "+d);

        //fractional part lost
        System.out.println("Long value "+l);

        //fractional part lost
        System.out.println("Int value "+i);
    }
}
```

Char to integer conversion

```
//Java program to illustrate incompatible data type for explicit type conversion
public class Test
{
  public static void main(String[] argv)
  {
    char ch = 'c';
    int num = 88;
    ch = num;
  }
}
```

String to integer

```
//Java program to illustrate incompatible data type for explicit type conversion
public class Test
{
  public static void main(String[] argv)
  {
    String age=“34”;
    int num = Integer.parseInt(age);
    System.out.println(num);
  }
}
```

Type Conversion

Data Type	Size	byte	short	int	long	float	double	char	boolean
byte	1	-							
short	2		-						
int	4			-					
long	8				-				
float	4					-			
double	8						-		
char	2							-	
boolean	I bit	-	-	-	-	-	-	-	-

Summary

We have discussed about

- Key points about Java
- Data types in Java
- Variables in Java
- Work with type casting in Java