HESHAN SANDARUWAN
200304512443

01.
Java data types are fundamental concepts in the Java programming language, and they serve several important purposes in programming:

Data Representation:
Memory Allocation
Type Safety
Performance
interoperability
Compile-Time Error Checking
Code Maintenance

data types:

Primitive Data Types
Reference Data Types

02.

primitive data types;

a. int type(32-bit)

```
int number = 42;
```

b. double type(64-bit floating-point numbers.)

```
double pi = 3.14159;
```

c. char type(represents a single character.)

```
char grade = 'A';
```

d. boolean type(either `true` or `false`.)

```
boolean isJavaFun = true;
```

Reference Data Types;

a. string type

```
String message = "Hello World!";
```

HESHAN SANDARUWAN
200304512443

03.

DATA TYPES	Data ranges
01.byte	-128 to 127(1-byte (8-bit))
02.short	-32768 to 32767(2-byte (16-bit))
03.int	-2.1 billion to 2.1 billion(4-byte (32-bit))
04.long	-9.2 quintillion to 9.2 quintillion(8-byte (64-bit))
05.float	4-byte (32-bit)
06.double	8-byte (64-bit)
07.char	2-byte (16-bit)
08.boolean	1-bit data

04.

a. boolean
f. char

05.

d. 0 to 216- 1

06.

while both `char` and `short` are 16-bit data types, they serve different purposes. `char` is used to represent characters and symbols from the Unicode character set, while `short` is used to store small signed integer values. The primary difference lies in their interpretation and how they are used in programming.

07.

a. "p"
b. 'h'
d. "Java"
e. true

HESHAN SANDARUWAN
200304512443

08.

In Java, the `int` data type is a 32-bit (4-byte) signed integer data type. This means that it is represented using 32 binary digits (bits) in computer memory.

The binary representation of an `int` is in two's complement form, which is a way to represent both positive and negative numbers using a fixed number of bits. The leftmost bit (the most significant bit) is the sign bit, where 0 indicates a positive number and 1 indicates a negative number. The remaining 31 bits are used to represent the magnitude of the number.

Decimal: 42

- Binary: 00101010 (31-bit magnitude, with a sign bit of 0 for positive)

And for a negative number, like -42:

- Decimal: -42
- Binary: 11010110 (31-bit magnitude, with a sign bit of 1 for negative)

09.

Number of Different Values = 2^n

10.

a. char firstLetterOfName = 'H';

b. Int numOfChildren = 2;

c. double weight = 75.5;

d. String spouseName = "hesh";

e. boolean isMarried = true;

HESHAN SANDARUWAN
200304512443

11.

the IDE highlights `String` differently because it wants to distinguish between primitive data types (like `int`, `double`, and `boolean`) and classes (like `String`). This helps developers quickly recognize when they are working with a class and its associated methods as opposed to a basic data type. This distinction is important for understanding how different data types behave in Java.

12.

Type casting in Java is the process of converting a value from one data type to another

Example of implicit type casting:

```
int smaller = 42;
double larger = smaller;
```

Also known as "Narrowing Conversion," this involves manually converting a larger data type into a smaller data type. Explicit casting is necessary because it can result in data loss or truncation, and Java wants to ensure that you are aware of this potential loss of information.

Example of explicit type casting:

```
double larger = 33.44;
int smaller = (int) larger;
```

HESHAN SANDARUWAN
200304512443

13.

1. Widening (Implicit) Type Conversion:

Widening, also known as implicit type conversion, is the process of converting a value from a smaller data type to a larger data type. This conversion is performed automatically by the Java compiler because it doesn't result in a loss of data.

- **Example**:

```
int smaller = 42;
double larger = smaller;
```

2. Narrowing (Explicit) Type Conversion:

Narrowing, also known as explicit type conversion, is the process of manually converting a value from a larger data type to a smaller data type. This conversion is performed by explicitly specifying the desired target data type. It may result in data loss or truncation.

Example:

```
double larger = 42.75;
Int smaller = (int) larger;
```

14.

In Java, a **keyword** is a reserved word that has a specific predefined meaning in the language. Keywords are an integral part of the Java programming language and are used to define the structure, behavior, and flow of Java programs.

0. **public**: This keyword is used to declare a class, method, or variable as accessible from anywhere in the Java program.
1. **class**: Used to declare a class, which is the blueprint for creating objects in Java. All Java code must be part of a class.
2. **static**: It is used to define a class-level variable or method, which means it belongs to the class rather than an instance of the class. Static members can be accessed without creating an object.
3. **void**: This keyword is used to indicate that a method does not return any value. It's the return type of a method that doesn't return a value.
4. **if**, **else**: Keywords used for conditional statements. `if` is used to specify a block of code to be executed if a condition is true, while `else` specifies a block to be executed if the condition is false.
5. **for**, **while**, **do**: Keywords used for loop control. `for` is used for traditional looping, `while` and `do` for creating while and do-while loops.

HESHAN SANDARUWAN
200304512443

6. **return**: Used to return a value from a method. It indicates the result of a method's execution.
7. **new**: Used to create a new instance of a class or to dynamically allocate memory for an object.
8. **final**: When used with a variable, it indicates that the variable's value cannot be changed. When used with a class, it indicates that the class cannot be extended (subclassed).
9. **try**, **catch**, **finally**: Keywords used for exception handling. `try` defines a block of code that may throw an exception, `catch` defines how to handle exceptions, and `finally` specifies code that should be executed whether an exception occurs or not.
10. **switch**, **case**: Keywords used to create a switch statement, which is used for multiple conditional branching based on a value.
11. **extends**: Used to create a subclass that inherits the properties and behaviors of a superclass in object-oriented programming.
12. **interface**: Used to declare an interface, which is a contract for classes to implement certain methods. Interfaces provide a form of multiple inheritance in Java.

15.

In Java, **identifiers** are names given to various program elements, such as variables, Methods, classes, interfaces, packages, and more. Identifiers serve as user-defined labels or names for these elements and are used to uniquely identify and access them within a program.

16.

In Java, a **"final" variable** is a variable that has been declared with the `final` modifier. A final variable, once assigned a value, cannot be changed or reassigned. It essentially becomes a constant, and its value is fixed for the entire duration of its existence. Final variables are also sometimes referred to as constants.

17.

Operator precedence and **associativity rules** in Java determine the order in which operators are evaluated when an expression contains multiple operators. Understanding these rules is crucial for writing code that behaves as expected, as they dictate the sequence in which operations are performed.

Common Operators and Their Precedence (from highest to lowest):

1. Postfix operators (e.g., `expr++`, `expr--`)
2. Unary operators (e.g., `++expr`, `--expr`, `+expr`, `-expr`, `~expr`, `!expr`)
3. Multiplicative operators (e.g., `*`, `/`, `%`)
4. Additive operators (e.g., `+`, `-`)
5. Shift operators (e.g., `<<`, `>>`, `>>>`)

HESHAN SANDARUWAN
200304512443

6. Relational operators (e.g., `<`, `>`, `<=`, `>=`, `instanceof`)
7. Equality operators (e.g., `==`, `!=`)
8. Bitwise AND operator (`&`)
9. Bitwise XOR operator (`^`)
10. Bitwise OR operator (`|`)
11. Logical AND operator (`&&`)
12. Logical OR operator (`||`)
13. Conditional (Ternary) operator (`? :`)
14. Assignment operators (e.g., `=`, `+=`, `-=`, `*=`, `/=`, `%=`)

18.

Unary Operators:

- Unary operators operate on a single operand.
- They are often used to perform operations on a single value or variable.

Ex;

`x = -5;` (negation)

- `y++;` (increment)
- `isTrue = !isFalse;` (logical NOT)

Binary Operators:

- Binary operators operate on two operands.
- They are used to perform operations that involve two values or variables.

Ex;

- `result = 5 + 3;` (addition)
- `isGreaterThan = x > y;` (greater than)
- `isBothTrue = a && b;` (logical AND)

Ternary Operator:

- The ternary operator is the only operator in this category.
- It operates on three operands.

Ex;

```
int max = (x > y) ? x : y;
```

19.

In Java, both the && and & operators are used for performing logical AND operations.

&& (Logical AND Operator):

- && is a **short-circuit logical AND operator**.
- Short-circuiting means that if the left-hand operand (expression) of && evaluates to `false`, the right-hand operand is not evaluated because the overall result will always be `false` regardless of the right-hand operand.

HESHAN SANDARUWAN
200304512443

ex;

if (x > 0 && y > 0)

& (Bitwise AND Operator):

- & is a **bitwise AND operator** when applied to integers, but it can also be used as a logical AND operator when applied to boolean operands.
- Unlike &&, & does not short-circuit the evaluation of the right-hand operand. It always evaluates both the left and right operands, whether the left-hand operand is `true` or `false`.

Ex;

if (x > 0 & y > 0)

20.

a. And(&)
b. OR (|)
c. XOR (^)
d. NOT (~)
e. Left Shift (<<)
f. Right Shift (>>)
g. Unsigned Right Shift (>>>)

21.

In Java, the ternary operator, also known as the conditional operator, is a shorthand way to write an `if-else` statement in a single line. It's called the "ternary" operator because it takes three operands: a conditional expression, a value to be returned if the condition is true, and a value to be returned if the condition is false.

```
int age = 20;
String message = (age >= 18) ? "You are an adult" : "You are
not an adult";
```

HESHAN SANDARUWAN
200304512443

22.

n = 6 k = 5

23.

46
1

24.

a. legal
b. illegal
c. legal
d. legal
e. legal
f. illegal
g. legal
h. legal

25.

a. legal
b. legal
c. illegal
d. legal
e. illegal
f. legal
g. illegal
h. illegal
I. illegal

26.

5
5
5
6
-6
6

HESHAN SANDARUWAN
200304512443

5
5
4
6
4
2
4 2
4
4
n

n * n =
16
n

27.

4
true
true
false
0
true
false
true
true
true

HESHAN SANDARUWAN
200304512443

28.

Int number	**Binary**	**Octal**	**Hexadecimal**
10	1010	12	A
16	10000	20	10
128	10000000	200	80
255	11111111	377	FF
32767	111111111111111	77777	7FFF
1	1	1	1
0	0	0	0
26	11010	32	1A
31	11111	37	1F

29.

a. 11110110
b. 10011100
c. 11000000
d. 11111111
e. 11111110
f. 10000000
g. 00000000
h. 10000001
I. 11100000

30.

a. u -= 1;
b. No such statement.
c. No such statement.
d. No such statement.
e. u %= v;

HESHAN SANDARUWAN
200304512443

31.

a. No such statement.
b. u -= (w % v – v);
c. v = v * u++;
d. u = u / (--u);
e. No such statement.

32.

a. legal
b. illegal
c. legal
d. illegal
e. legal
f. legal

33.

line

1. compile error(cannot be directly assigned to `b3`)
2. valid
3. valid
4. valid
5. valid
6. compilation error
7. valid
8. assigned the value `30`
9. valid
10. valid
11.valid
12. valid
13.valid

HESHAN SANDARUWAN
200304512443

34.

a. `l = 2147483647;`
c. `l = 0xabcd;`
e. `l = 0101010110L;`

35.

a. avg = (double) tot/10;
b. avg = tot/(double)10;

36.

a. 3
b. -3
c. 3
d. -3
e. 3
f. -3
g. 3

37.

b. 4.0 4.5 4.5 5.0

38.

e. x=Byte.MIN_VALUE;

39.

a d c e

HESHAN SANDARUWAN
200304512443

40.

a. legal
b. illegal
c. legal
d. legal
e. legal

41.

a. legal
b. illegal
c. legal
d. illegal

42.

a. 17
b. -10
c. -17
d. -3
e. 7
f. -3

43.

- 100
- -100
- 100
- 200
- -200
- 0

HESHAN SANDARUWAN
200304512443

44.

100
101
104
104

45.

```
101 100
102 101
103 102
```

46.

101 101
102 102
103 103

47.

100
100
100
101
102
103

48.

The code segment `(x++)++;` causes a compilation error because the post-increment operator `x++` returns a value, and you cannot increment the result of a value.

49.

300
306

402 101

50.

a. 21
b. 22
c. 26
d. 27
e. 31
f. 33
g. 38
h. 42
i. 44
j. 48

51.

a. x = 30
b. x = -10
c. x = 31
d. x = 30
e. x = 31
f. x = 30
g. x = 32
h. x = 30

52.

a. 3
b. 0
c. 10
d. 0.0
e. 0.5

HESHAN SANDARUWAN
200304512443

53.

a. x = 6
b. x = 4
c. x = 1
d. x = 1
e. x = 7

54.

a. a = 106
b. a = 106
c. a = 11
d. a = 400

55.

12 - 4 * 2 : 4
(12 - 4) * 2 : 16
12 - (4 * 2) : 4

56.

a.

```
public class Demo1 {

  public static void main(String[] args) {
        final double ratio = 1.8;
          int b, c = 10;
             int d = c + 5;
           c = 10; // Corrected from "c = 10c:"
    b = d / c;

          System.out.println("b = " + b);
         System.out.println("c = " + c);
        System.out.println("ratio = " + (b / d));
  }
```

HESHAN SANDARUWAN
200304512443

b.

```
public class Demo2 {

 public static void main(String[] param) {
    final char myChar = '\7777';
    final int TEN;
    int a, b, c;
    TEN = a + 5;
    b = a + myChar;
    c = a + 5 * TEN;
    b += c;
    System.out.print("a + a");
    System.out.print("myChar + myChar");
    System.out.print("TEN = " + TEN);
  }
}
```

c.

```
        public class Demo3 {

  public static void main(String[] args) {
    final int OFFSET = 32;
    int b, c = 7, d;
    d = c + 5;
    c = d - b;
    b = d / c;
    c = b % (OFFSET + 1);
    d += b;
    System.out.println("B = " + b);
    System.out.println("C = " + c);
    System.out.println("D = " + d);
  }
}
```

HESHAN SANDARUWAN
200304512443

d.

```
public class Demo4 {

   public static void main(String[] args) {
      final char myChar = 'A';
      final int SIXTY_SIX = 'B';
      int a, b, c;
      a = 10;
      b = a + myChar;
      c = a + 5 * SIXTY_SIX;
      a = (b + c)++;
      System.out.print("a: " + a);
      System.out.print("b: " + b);
      System.out.print("(a+b) : " + (a + b));
   }
}
```

57.

1. 37
2. 7
3. 8
4. -13.34999999999999
5. 5.625
6. 237.0
7. 15.0
8. 6
9. -10
10. 10
11. -9
12. 4
13. 4
14. 2

HESHAN SANDARUWAN
200304512443

58.

2 3 4 5 10
14 12 9 5 24
5 5 5 5 5

59.

false
true
true
true false true true false

60.

false
false
true
false

61.

```
import java.util.*;

public class Main{
public static void main(String args[]){
Scanner input = new Scanner(System.in);

System.out.print("Input value celcius :");
float C =input.nextFloat();

float F = (C * 9 / 5) + 32;


System.out.println("(Temperature in degrees Fahrenheit ="+F);
}}
```

HESHAN SANDARUWAN
200304512443

62.

```
import java.util.*;

public class Main {
   public static void main(String[] args) {
      Scanner input = new Scanner(System.in);

      System.out.print("Enter weight in kilograms: ");
      double weight = input.nextDouble();

      System.out.print("Enter height in meters: ");
      double height = input.nextDouble();


      double bmi = weight / (height * height);

      System.out.println("BMI: " + bmi);



   }
}
```

63.

```
import java.util.*;

public class Main {
   public static void main(String[] args) {
      Scanner input = new Scanner(System.in);

      System.out.print("Enter meters: ");
      double Meters = input.nextDouble();

      System.out.print("Enter seconds: ");
      double Seconds = input.nextDouble();
```

HESHAN SANDARUWAN
200304512443

```
        double speedInMetersPerHour = (Meters / Seconds) * 3600;

        System.out.println("Speed in meters per hour: " +
         speedInMetersPerHour + " m/hour");

    }
}
```

64.

```
import java.util.*;

public class Main{
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter num: ");
        int number = input.nextInt();

        if (number < 100 || number > 999) {
            System.out.println("not");
        } else {
            int s1 = number % 10;
            int s2 = (number / 10) % 10;
            int s3 = number / 100;

            int sum = s1 + s2 + s3;

            System.out.println("Sum: " + sum);
        }


    }
}
```

HESHAN SANDARUWAN
200304512443

65.

```
import java.util.*;

public class Main{
   public static void main(String[] args) {
      Scanner input = new Scanner(System.in);

      System.out.print("Enter obtained marks: ");
      double Marks = input.nextDouble();

      System.out.print("Enter maximum marks: ");
      double maximumMarks = input.nextDouble();


      double percentage = (Marks / maximumMarks) * 100;

      System.out.println("Percentage: " + percentage + "%");



   }
}
```

66.

```
import java.util.*;

public class Main {
   public static void main(String[] args) {
      Scanner input = new Scanner(System.in);

      System.out.print("Input seconds: ");
      int Seconds = input.nextInt();

      int Minute = 60;
```

HESHAN SANDARUWAN
200304512443

```
        int Hour = 60 * Minute;
        int Day = 24 * Hour;

        int days = Seconds / Day;
        int Seconds1 = Seconds % Day;
        int hours = Seconds / Hour;
        Seconds %= Hour;
        int minutes = Seconds / Minute;
      int seconds = Seconds % Minute;

        System.out.printf("%02d:%02d:%02d:%02d%n",days, hours,
minutes, seconds);


    }
}
```

67.

```
        import java.util.*;

public class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter total sales amount: ");
        double totalSalesAmount = input.nextDouble();

        System.out.print("Enter commission rate ): ");
        double commissionRate = input.nextDouble();


        double commission = totalSalesAmount * commissionRate;

        System.out.println("Commission earned: " + commission);

        }}
```

HESHAN SANDARUWAN
200304512443

68.

```
import java.util.*;

public class Main {
   public static void main(String[] args) {
      Scanner input = new Scanner(System.in);

      System.out.print("Enter the value of 'a': ");
      double a = input.nextDouble();

      System.out.print("Enter the value of 'b': ");
      double b = input.nextDouble();


         double x = -b / a;
         System.out.println("Solution: x = " + x);



   }
}
```

69.

```
import java.util.*;

public class Main {
   public static void main(String[] args) {
      Scanner input = new Scanner(System.in);


      double sum = 0;

      for (int i = 1; i <= 5; i++) {
         System.out.print("Enter temperature " + i + ": ");
         double temperature = input.nextDouble();
         sum = sum +temperature;
```

HESHAN SANDARUWAN
200304512443

```
        }

        double avg = sum / 5;

        System.out.println("Average temperature: " + avg );


    }
}
```

70.

```
         import java.util.*;

public class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter the number of eggs: ");
        int Eggs = input.nextInt();

        int dozenEggs = Eggs / 12;
        int grossEggs = Eggs / 144;
        int extraEggs = Eggs % 12;

        System.out.println("You have:");
        System.out.println("Dozen eggs: " + dozenEggs);
        System.out.println("Gross: " + grossEggs);
        System.out.println("Extra eggs: " + extraEggs);


    }
}
```

HESHAN SANDARUWAN
200304512443

71.

```
import java.util.*;

public class Main {
   public static void main(String[] args) {
      Scanner input = new Scanner(System.in);

      System.out.print("Enter salary: ");
      double Salary = input.nextDouble();


      double annualSalary = Salary * 12;


      double bonus = 1000 + 0.02 * (annualSalary - 7000);

      System.out.println("Bonus " + bonus);


   }
}
```

72.

```
import java.util.*;

public class Main {
   public static void main(String[] args) {
      Scanner input = new Scanner(System.in);

      System.out.print("Enter  runs scored: ");
      int runs =input.nextInt();

      System.out.print("Enter  innings played: ");
      int innings = input.nextInt();
```

HESHAN SANDARUWAN
200304512443

```
        System.out.print("Enter  player remained not out: ");
        int notOut = input.nextInt();

        if (innings - notOut == 0) {
            System.out.println("The player's average cannot ");
        } else {
            double average = (double) runs / (innings - notOut);
            System.out.println("Batting average of the player: " + average);
        }

    }
}
```

73.

```
        import java.util.*;

public class Main{
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter radius of the circle: ");
        double radius = input.nextDouble();

        double area = 22/7.0 * radius * radius;

        double perimeter = 2.0 * 22/7 * radius;

        System.out.println("Area circle: " + area);
        System.out.println("Perimeter circle: " + perimeter);

    }
}
```

HESHAN SANDARUWAN
200304512443

74.

a. 4 / 3 * (r + 34) - 9 * (a + b * c) + (3 + d * (2 + a)) / (a + b * d)

b. 5.5 * (r + 2.5) / (2.5 + t)

75.

```
import java.util.Scanner;

public class Main{
   public static void main(String[] args) {
      Scanner input = new Scanner(System.in);

      System.out.print("Enter marks obtained in subject 1: ");
      double x1 = input.nextDouble();

      System.out.print("Enter marks obtained in subject 2: ");
      double x2 = input.nextDouble();

      System.out.print("Enter marks obtained in subject 3: ");
      double x3 = input.nextDouble();

      System.out.print("Enter mean mark of subject 1: ");
      double m1 = input.nextDouble();

      System.out.print("Enter mean mark of subject 2: ");
      double m2 = input.nextDouble();

      System.out.print("Enter mean mark of subject 3: ");
      double m3 = input.nextDouble();

      System.out.print("Enter standard deviation mark of subject 1: ");
      double sd1 = input.nextDouble();

      System.out.print("Enter standard deviation mark of subject 2: ");
      double sd2 = input.nextDouble();
```

HESHAN SANDARUWAN
200304512443

```
        System.out.print("Enter standard deviation mark of subject 3: ");
        double sd3 = input.nextDouble();
  double zScore = ((x1 - m1) / sd1 + (x2 - m2) / sd2 + (x3 - m3) / sd3) /3.0;

        System.out.println("Z-score of the student: " + zScore);


    }
}
```

76.

```
        import java.util.*;

public class Main{
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter x1: ");
        double x1 = input.nextDouble();

        System.out.print("Enter y1: ");
        double y1 = input.nextDouble();

        System.out.print("Enter x2: ");
        double x2 = input.nextDouble();

        System.out.print("Enter y2: ");
        double y2 = input.nextDouble();


        double distance = Math.sqrt(Math.pow(x2 - x1, 2) + Math.pow(y2 -
y1, 2));

        System.out.println("Distance between the points: " + distance);
    }]
```

HESHAN SANDARUWAN
200304512443

77.

```
import java.util.*;

public class Main {
   public static void main(String[] args) {
      Scanner input = new Scanner(System.in);

      System.out.print("Enter the principal amount : ");
      double principal = input.nextDouble();

      System.out.print("Enter the annual interest rate : ");
      double annualInterestRate = input.nextDouble();

      System.out.print("Enter the time period in years : ");
      double timeInYears = input.nextDouble();


      double compoundInterest = principal * (Math.pow(1 +
annualInterestRate / 100, timeInYears) - 1);

      System.out.println("Compound Interest: " + compoundInterest);

   }
}
```

HESHAN SANDARUWAN
200304512443
78.

```
        import java.util.

public class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter the coefficient: ");
        double a = input.nextDouble();

        System.out.print("Enter the coefficient: ");
        double b = input.nextDouble();

        System.out.print("Enter the coefficient  ");
        double c = input.nextDouble();

        double discriminant = b * b - 4 * a * c;

        if (discriminant > 0) {
            double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
            double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);

            System.out.println("The roots are real and distinct:");
            System.out.println("Root 1: " + root1);
            System.out.println("Root 2: " + root2);
        } else if (discriminant == 0) {
            double root = -b / (2 * a);
            System.out.println("The root is real and equal: " + root);
        } else {
            System.out.println("The roots are complex and not real.");
        }


    }
}
```