Answer the following

1. List primitive data types with respect to size and explain them with example.

In Java, primitive data types are the most basic data types. Below are the primitive data types in Java along with their sizes:

Data Type	Size	Description	Example
byte	1 byte	Stores whole numbers from -128 to 127.	byte b = 10;
short	2 bytes	Stores whole numbers from -32,768 to 32,767.	short s = 1000;
int	4 bytes	Stores whole numbers from -2^31 to 2^31-1.	int i = 100000;
long	8 bytes	Stores whole numbers from -2^63 to 2^63-1.	long l = 100000000L;
float	4 bytes	Stores fractional numbers, sufficient for storing 6-7 decimal digits.	float f = 5.75f;
double	8 bytes	Stores fractional numbers, sufficient for storing 15 decimal digits.	double d = 19.99;
char	2 bytes	Stores a single character/letter or ASCII values.	char c = 'A';
boolean	1 bit	Stores `true` or `false` values.	boolean flag = true;

2. Explain type casting concepts of widening and narrowing.

Type casting in Java refers to converting one data type into another. There are two types of casting:

- **Widening (Automatic/Implicit Casting)**: This happens when a smaller data type is converted into a larger one. This type of casting is automatic and does not require explicit syntax.
  - **Example**:

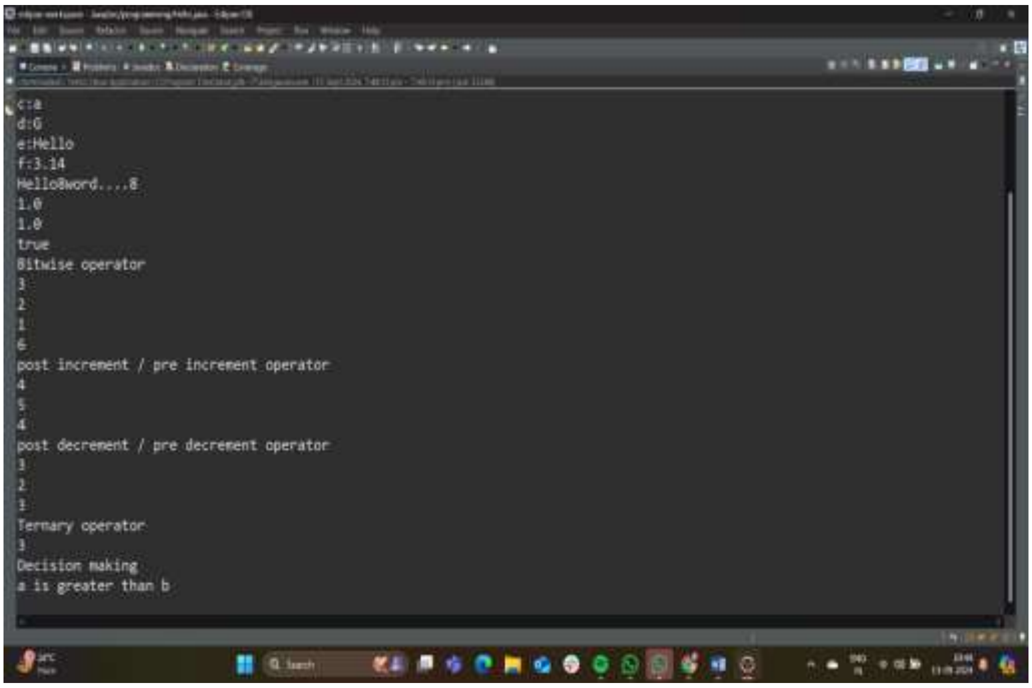
```
int num = 100;
double d = num;  // Widening from int to double
```

- **Narrowing (Explicit Casting)**: This is when a larger data type is converted into a smaller data type. It requires explicit casting because there is a potential loss of data.
  - **Example**:

```
double d = 9.78;
int num = (int) d;  // Narrowing from double to int
```

```
package programming;

public class Hello {

	public static void main(String[] args) {
			// variable declaring and initialize
			int a = 8;
			byte b;
			//type conversion(Narrowing Type)
			b = (byte) a;
			byte g = 8;
			int h;
			float x = 3;
			float y = 2;
			int ans1;
			ans1 = (int) (x/y);
			//type conversion(Widening Type)
			h = g;
			// other data type
			char c = 'a';
			char d = 71;
			String e = "Hello";
			double f = 3.14;
			int i = 3;
			int j = 2;
			float ans;
			ans = i/j;
			double x1 = 3.5;
			double y2 = 2.5;
			double ans2 = x1 % y2;
			// Simple Hello World print
			System.out.println("Hello word....");
			// variable print
			System.out.println("Hello"+ a +"word....");
			//type conversion(Narrowing Type)
			System.out.println("Hello"+ a +"word...." + b);
			System.out.println(ans1);
			//variables print
			System.out.println("a:"+ a +"\nb:" + b + "\nc:" + c + "\nd:" + d +
"\ne:" + e + "\nf:" + f);
			//type conversion(Widening Type)
			System.out.println("Hello"+ g +"word...." + h);
			System.out.println(ans);
			System.out.println(ans2);

			int x2 = 3;
			int y1 = 2;
			boolean ans3 = x2>y1;

			System.out.println(ans3);
```

```
                System.out.println("Bitwise operator");
                // Bitwise operator
                int a1 = 3;
                int b1 = 2;
                System.out.println(a1|b1);
                System.out.println(a1&b1);
                System.out.println(a1>>1);
                System.out.println(a1<<1);

                System.out.println("post increment / pre increment operator");
                int a2 = 3;
                a2++;
                System.out.println(a2);
                ++a2;
                System.out.println(a2);
                int a3 = 3;
                ++a3;
                System.out.println(a3);

                System.out.println("post decrement / pre decrement operator");
                int b2 = 4;
                --b2;
                System.out.println(b2);
                b2--;
                System.out.println(b2);
                int b3 = 4;
                b3--;
                System.out.println(b3);

                System.out.println("Ternary operator");
                int a4 = 3;
                int b4 = 2;
                int c1 = (a4>b4)?a4:b4;
                System.out.println(c1);


                System.out.println("Decision making");
                int a5=3;
                int b5=2;
                if(a5>b5)
                {
                        System.out.println("a is greater than b");
                }
                else {
                        System.out.println("b is greater than a");
                }

        }
}
```



```
c:a
d:G
e:Hello
f:3.14
Hello8word....8
1.0
1.0
true
Bitwise operator
3
2
1
6
post increment / pre increment operator
4
5
4
post decrement / pre decrement operator
3
2
3
Ternary operator
3
Decision making
a is greater than b
```

rd

3. What are Comments and their purpose in Java. State the types of comments in Java.

Comments in Java are annotations that help developers understand the code. The compiler ignores comments, meaning they are not executed. The main purpose of comments is to explain the code for better readability and maintainability.

Java supports three types of comments:

Single-line Comments: Used for short explanations, starting with //.

Example:

```
// This is a single-line comment
int x = 5;
```

Multi-line Comments: Used for longer explanations, enclosed between /* and */.

Example:

```
/* This is a multi-line comment
   that spans multiple lines. */
int y = 10;
```

Javadoc Comments: Special comments used to generate documentation, starting with /** and ending with */.

Example:

```
/**
 * This method returns the sum of two numbers.
 * @param a First number
 * @param b Second number
 * @return Sum of a and b
 */
public int add(int a, int b) {
    return a + b;
}
```