Type Casting in Java

Detailed Notes

Type casting changes a variable’s data type. It helps convert values when you need to assign a variable of one type to

Implicit (Widening) Casting

Happens automatically; converting smaller to larger types.

No risk of data loss.

Examples: int long, float double

Explicit (Narrowing) Casting

Required when converting larger to smaller types.

Potential data loss; manual syntax needed.

Syntax: (targetType) value

Casting Between Primitive and Reference

Primitive to primitive can be casted.

Casting from primitives to objects (like String) uses specific methods, not direct casting.

Sample Programs

java
```java
public class CastingDemo {
    public static void main(String[] args) {
        int i = 100;
        double d = i; // Implicit: int to double
        System.out.println("Implicit casting double: " + d);

        double x = 9.78;
        int y = (int) x; // Explicit: double to int
        System.out.println("Explicit casting int: " + y);

        String strNum = "50";
        int num = Integer.parseInt(strNum);  // String to int conversion
        System.out.println("String to int: " + num);

        String floatStr = String.valueOf(24.56f); // float to String
        System.out.println("Float to String: " + floatStr);
    }
}
```

Interview Questions and Answers: Type Casting

What is type casting and why is it needed in Java?

Type casting converts data from one type to another, allowing compatibility between different types during assignm
oesn’t allow implicit conversions that could cause errors or data loss.

What is the difference between widening and narrowing casting?

Widening casting automatically converts a smaller datatype to a larger one without losing info (safe). Narrowing requ
a smaller datatype, potentially causing truncation or precision loss.

Can you give an example of widening casting in Java?

When you assign an int to a double, Java automatically widens the 32-bit integer to a 64-bit double:

java
```java
int num = 42;
double d = num;  // automatic widening to 42.0
```

What is the syntax and use case of narrowing casting?

Explicit casting is done as follows:

java
```
double d = 9.99;
int i = (int) d; // i becomes 9, fractional part lost
```
Use it when you need to convert a larger numeric type to a smaller one but accept data loss.

How can you convert a String value to an int in Java?
Use parsing methods:

java
```
String s = "123";
int num = Integer.parseInt(s);
```
This converts the numeric string into an integer. Errors occur if string isn't a valid number.

Can you cast primitive types to Strings?
No direct casting; you use methods like String.valueOf() or concatenation, e.g.:

java
```
int n = 50;
String str = String.valueOf(n);
```
What happens if you fail to cast properly?
Incorrect casting can cause compile-time errors or runtime exceptions (ClassCastException, NumberFormatExceptio

Is it safe to cast double to float implicitly?
No, because double has higher precision and size; you must explicitly cast it, accepting the potential loss of precisior

Why does Java not allow automatic casting from double to int?
Because it can cause loss of decimal information and may truncate the value, Java forces explicit casting to alert the

How are characters handled in casting?
You can cast a char to its corresponding integer Unicode value and vice versa:

java
```
char c = 'A';
int code = (int) c;  // 65
```
Assignments: Type Casting
Write a program demonstrating all possible widening casts between primitive types.

Write a method that takes a double array and casts each element explicitly to int, printing results.

Develop a program converting string inputs to appropriate numeric types with error handling.

Show examples where improper casting causes exceptions and how to catch them.

Implement a calculator converting user input strings into numbers and performing arithmetic operations.