Java Programming Assignment

Section 1: Java Data Types

1. What are the different primitive data types available in Java?

ANS: Java has 8 primitive data types:

byte – 1 byte, stores small integers (-128 to 127)

short – 2 bytes, stores medium integers (-32,768 to 32,767)

int – 4 bytes, default integer type

long – 8 bytes, stores large integers

float – 4 bytes, single-precision decimal

double – 8 bytes, double-precision decimal

char – 2 bytes, stores a single character (Unicode)

boolean – 1 bit, stores true or false

2. Explain the difference between primitive and non-primitive data types in Java.
ANS:

Primitive	Non-Primitive
Fixed size means memory used is predefined and does not change.	Variable size means memory usage depends on the object's content or structure.
Examples: int, char, boolean, byte, short, long, float, double.	Examples: String, Array, Class, Interface, Object.
Stored directly in stack memory (faster access).	Stores a reference (address) in stack, actual object is in heap memory.
Cannot be null (default values are assigned).	Can be null if no object is assigned.

3. Write a Java program that demonstrates the use of all primitive data types.

ANS:
```java
package Day_1;

public class PrimitiveDemo {
    public static void main(String[] args) {
        byte b = 100;
        short s = 2000;
        int i = 50000;
        long l = 900000L;
        float f = 5.75f;
        double d = 19.99;
        char c = 'A';
        boolean bool = true;
        System.out.println("byte: " + b);
        System.out.println("short: " + s);
        System.out.println("int: " + i);
        System.out.println("long: " + l);
        System.out.println("float: " + f);
        System.out.println("double: " + d);
        System.out.println("char: " + c);
        System.out.println("boolean: " + bool);
    }
}
```

```
OUTPUT:
 short: 2000
 int: 50000
 long: 900000
 float: 5.75
 double: 19.99
 char: A
 boolean: true
```

4. What is type casting? Provide an example of implicit and explicit casting in Java.

ANS: Type casting means converting one data type into another.

- Widening (Implicit) – small type → large type
  int num = 10;
  double val = num; // int to double
- Narrowing (Explicit) – large type → small type
  double pi = 3.14;
  int intPi = (int) pi; // double to int

5. What is the default value of each primitive data type in Java?

ANS:

Data Type	Default Value
byte, short, int, long	0
float, double	0.0
char	'\u0000' (null character)
boolean	false

Section 2: Java Control Statements

1. What are control statements in Java? List the types with examples.

ANS: Control statements decide how the flow of execution moves in a program based on conditions, loops, or jumps.

Types of Control Statements:

1. Conditional Statements – Make decisions
   - if, if-else, if-else-if, nested if, switch
2. Looping Statements – Repeat code
   - for, while, do-while

3. Jump Statements – Change normal flow
    - break, continue, return

2. Write a Java program to demonstrate the use of if-else and switch-case statements.

ANS:
```java
package Day_1;

public class ControlDemo {
    public static void main(String[] args) {
        int num = 5;

        if (num % 2 == 0) {
            System.out.println(num + " is Even");
        } else {
            System.out.println(num + " is Odd");
        }

        int day = 3;
        switch (day) {
            case 1: System.out.println("Monday"); break;
            case 2: System.out.println("Tuesday"); break;
            case 3: System.out.println("Wednesday"); break;
            default: System.out.println("Invalid day");
        }
    }
}
```

Output:

```
5 is Odd
Wednesday
```

3. What is the difference between break and continue statements?

ANS:

break	**continue**
Exits the loop or switch immediately.	Skips current loop iteration and moves to the next one.
Example: stop loop when condition met.	Example: skip printing a specific value.

4. Write a Java program to print even numbers between 1 to 50 using a for loop.

ANS:
```java
package Day_1;

public class EvenNumbers {
    public static void main(String[] args) {
        for (int i = 1; i <= 50; i++) {
            if (i % 2 == 0) {
                System.out.print(i + " ");
            }
        }
    }
}
```

Output:

```
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50
```

5. Explain the differences between while and do-while loops with examples.

ANS:

while	**do-while**
Condition is checked before executing loop body.	Condition is checked after executing loop body.
May not run at all if condition is false initially.	Runs at least once regardless of condition.

Section 3: Java Keywords and Operators

1. What are keywords in Java? List 10 commonly used keywords.

ANS: Keywords are reserved words in Java with predefined meaning.

- Cannot be used as variable, class, or method names.
- Examples: int, class, if, else, static, final, void, public, this, return

2. Explain the purpose of the following keywords: static, final, this, super.

ANS: static- Belongs to class, not to objects; can be accessed without creating an object.

Final-Used to declare constants, prevent method overriding or inheritance.

This-Refers to the current object of the class.

Super-Refers to the parent class object, used to call parent methods or constructors.

3. What are the types of operators in Java?

ANS:

- **Arithmetic** – +, -, *, /, %
- **Relational** – ==, !=, >, <, >=, <=
- **Logical** – &&, ||, !
- **Assignment** – =, +=, -=, *= etc.
- **Unary** – ++, --, +, -
- **Bitwise** – &, |, ^, ~
- **Shift** – <<, >>, >>>

4. Write a Java program demonstrating the use of arithmetic, relational, and logical operators.

ANS:
```
package Day_1;

public class OperatorDemo {
    public static void main(String[] args) {
        int a = 10, b = 5;

        // Arithmetic
        System.out.println("Sum: " + (a + b));

        // Relational
        System.out.println("a > b: " + (a > b));

        // Logical
        System.out.println("a > 0 && b > 0: " + (a > 0 && b > 0));
    }
}
```
Output:
```
Sum: 15
a > b: true
a > 0 && b > 0: true
```

5. What is operator precedence? How does it affect the outcome of expressions?

ANS: Operator precedence decides which operation is done first when an expression has multiple operators.

Example:

int result = 10 + 5 * 2;

System.out.println(result);

Additional Questions

Java Data Types

6. What is the size and range of each primitive data type in Java?

ANS:

Type	Size	Range
byte	1 byte	-128 to 127
short	2 bytes	-32,768 to 32,767
int	4 bytes	-2^{31} to $2^{31} - 1$
long	8 bytes	-2^{63} to $2^{63} - 1$
float	4 bytes	~±3.4e38
double	8 bytes	~±1.7e308
char	2 bytes	Unicode 0 to 65,535
Boolean	1 bit	true / false

7. How does Java handle overflow and underflow with numeric types?

ANS: When a numeric variable exceeds its maximum value, it overflows and wraps around to the minimum value. Similarly, going below the minimum value causes underflow, wrapping to the maximum value.

8. Write a program to convert a double value to an int without data loss.

ANS:
```
package Day_1;

public class TypeCastExample {
    public static void main(String[] args) {
        double d = 12345.0;
        int i = (int) d;
        System.out.println(i);
    }
}
12345
```

9. What is the difference between char and String in Java?

ANS:

char	**String**
Stores a single character.	Stores multiple characters (sequence).
Primitive data type.	Non-primitive (object of String class).
Example: 'A'	Example: "Hello"

10. Explain wrapper classes and their use in Java.

ANS: Wrapper classes allow primitive types to be used as objects. For example, Integer wraps an int, and Double wraps a double. They are often used with collections and for type conversions.

Java Control Statements

6. Write a Java program using nested if statements.

ANS:
```
package Day_1;

public class Nested_if {

	public static void main(String[] args) {
		int age=20;
		boolean hasID = true;
		if (age >= 18) {
		    if (hasID) {
		        System.out.println("Entry allowed");
		    } else {
		        System.out.println("ID required");
		    }
		} else {
```

```
                System.out.println("Not allowed");
            }


        }

}
```

Output:

Entry allowed

7. Write a Java program to display the multiplication table of a number using a loop.

ANS:

```
package Day_1;

public class multiplication {

        public static void main(String[] args) {
                int num = 5;
                for (int i = 1; i <= 10; i++) {
                    System.out.println(num + " x " + i + " = " + (num * i));
                }


        }

}
```

Output:

```
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

8. Compare and contrast for, while, and do-while loops.

ANS:

for	**while**	**do-while**
Entry-controlled	Entry-controlled	Exit-controlled
Best when number of iterations known	When condition may change in loop	Runs at least once
Syntax compact	Syntax simple	Slightly longer syntax

9. Write a program that uses a switch-case to simulate a basic calculator.

ANS:

```
package Day_1;

public class switch_case {
        public static void main(String[] args) {
                int a = 10, b = 5;
                char op = '+';
                switch (op) {
                    case '+': System.out.println(a + b); break;
                    case '-': System.out.println(a - b); break;
                    case '*': System.out.println(a * b); break;
                    case '/': System.out.println(a / b); break;
                    default: System.out.println("Invalid operator");
                }


        }

}
Output: 15
```

Java Keywords and Operators

1. What is the use of the `instanceof` keyword in Java?

ANS: The instanceof keyword checks if an object belongs to a specific class.

String s = "Hello";

System.out.println(s instanceof String);

2. Explain the difference between `==` and `.equals()` in Java.

ANS: == compares memory addresses for objects, while .equals() checks content equality (if overridden). For strings, "abc".equals("abc") returns true, but == might not.

3. Write a program using the ternary operator.

ANS:
```
package Day_1;

public class ternary_operator {

        public static void main(String[] args) {
                int num = 5;
                String result = (num % 2 == 0) ? "Even" : "Odd";
                System.out.println(result);


        }

}
Output: Odd
```

4. What is the use of `this` and `super` in method overriding?

ANS: this refers to the current object, while super refers to the parent class. In overriding, super can call the parent's method

5. Explain bitwise operators with examples.

ANS:
```
package Day_1;

public class Bitwise {

	public static void main(String[] args) {
		int a = 5;
		int b = 3;
		System.out.println(a & b);
		System.out.println(a | b);
		System.out.println(a ^ b);
		System.out.println(~a);


	}

}
```
Output:
```
1
7
6
-6
```