

Student Name \_\_\_\_\_

- The compiler uses information about the type of a variable to determine how much memory to allocate for the variable and to check that it is used correctly.
- Creating a class creates a new type whose name is the name of the class.
- The `new` operation is used to create reference - type variables.
- Classes have special methods called constructors to construct objects.
- An expression of the form `new ClassName (arguments)` constructs a `ClassName` object, using the specified arguments to initialize it, and returns the address of this object. Storing this address in a variable of type `ClassName` as in  

```
ClassName variable = new ClassName (arguments);
```

makes `variable` a handle for this object by which it can be accessed and used.
- Integer literals that begin with `0` are interpreted as hexadecimal (base - 16) integers if they begin with `0x` and otherwise as octal (base - 8) values.
- In Java, real values can be represented in fixed - point notation - `m.n` - or in floating - point notation - `xEn` or `xen` where `x` is fixed - point form and `n` is an integer.
- All real literals are treated as of type `double`.
- If `a` and `b` are both integers with `b ≠ 0`, `a / b` gives the integer quotient when `a` is divided by `b`, and `a % b` gives the remainder. If `a` or `b` is real, real division is used for `a / b` and `a % b` results in an error.
- In an expression, `*`, `/`, and `%` have higher precedence than (i.e., are applied before) `+` and `-`.
- Associativity determines whether equal - priority operators are applied from left to right or from right to left.
- Parentheses can be used to change the usual order of evaluation in an expression.
- `static` (or `class`) methods are invoked by sending a message to a `class`; `instance` methods are invoked by sending a message to an object (i.e., an instance of a class).
- Inside a class definition, a method not preceded by the keyword `static` is an instance method; a method preceded by `static` is a static (or class) method.
- Java's `Math` class (in its `java.lang` package) contains constants and methods for several of the familiar mathematical constants and functions.

Student Name \_\_\_\_\_

- `float`, `double`, `byte`, `short`, `integer`, `long`, and `character` are wrapper classes that provide added capabilities for the corresponding primitive types; one important use is for type conversion. For example, two useful type - conversion methods in the `Integer` class are `toString()` and `parseInt()`.
- Classes `BigInteger` and `BigDecimal` in the `java.math` package are useful for processing values too large to be represented by the `long` and `double` types, respectively.
- In numeric expressions that contain mixed types, narrower numeric values are promoted automatically to wider ones and integers to reals.
- Type - casting can be used to convert one numeric primitive type to another.
- An assignment statement is a replacement statement: `a = b` replaces the value of `a` with the value of `b`; `b = a` replaces the value of `b` with the value of `a`.
- Like other binary operators, the assignment operator `=` produces a value. In an assignment expression `a = b`, `=` is an operator with operands `a` and `b`. In addition to replacing the value of `a` with the value of `b`, it also produces this value as a result.
- A prefix increment expression `++variable` increments the value of the variable by 1 and produces this new value as the result. A postfix increment expression `variable++` increments the value of the variable by 1 but produces its original value as the result. Decrement expressions behave in a similar fashion.
- In an expression of the form `a + b`, where `a` or `b` is of type `String` and the other is of a primitive type, the non - `String` operand will be converted to type `String` and treated as concatenation.
- For `str` of type `String` and `i` an integer, `str.charAt(i)` is the character in position `i` of `str` (counting from 0).