Lecture #2

Review & Tips

What we have learned so far

- Types & Variables
- Operators
- Type conversions & casting
- Methods & parameters
- *If* statement

Types

Kinds of values that can be stored and manipulated.

boolean: Truth value (**true** or **false**).

int: Integer (0, 1, -47).

double: Real number (3.14, 1.0, -2.1).

String: Text (“hello”, “example”).

Variables

Named location that stores a value

Example:

```
String a = “a”;
String b = “letter b”;
a = “letter a”;
String c = a + “ and “ + b;
```

Mismatched Types

Java verifies that types always match:

```
String five = 5;   // ERROR!
```

```
test.java.2: incompatible types  found: int
required: java.lang.String
String five = 5;
```

Conversion by method

int to String:

```
String five = 5; // ERROR!
String five = Integer.toString (5);
String five = “” + 5;  // five = “5”
```

String to int:

```
int foo = “18”; // ERROR!
int foo = Integer.parseInt (“18”);
```

Conversion by casting

```
int a = 2;      // a = 2
double a = 2;   // a = 2.0 (Implicit)

int a = 18.7;         // ERROR
int a = (int)18.7;       // a = 18

double a = 2/3;      // a = 0.0
double a = (double)2/3;   // a = 0.6666…
```

Order of Operations

Precedence like math, left to right

Right hand side of = evaluated first

Parenthesis increase precedence

```
double x = 3 / 2 + 1; // x = 2.0
double y = 3 / (2 + 1); // y = 1.0
```

if statement

```
if (CONDITION) {
    STATEMENTS
}
```

else

```
if (CONDITION) {
    STATEMENTS
} else {
    STATEMENTS
}
```

else if

```
if (CONDITION) {
    STATEMENTS
} else if (CONDITION) {
    STATEMENTS
} else if (CONDITION) {
    STATEMENTS
} else {
    STATEMENTS
}
```

```
class Foo {
        public static void main(String[] arguments) {

                Scanner input = new Scanner(System.in);
                int x = input.nextInt();

                if (x > 5) {
                   System.out.println(x + " is > 5");
                } else if (x == 5) {
                   System.out.println(x + " equals 5");
                } else {
                   System.out.println(x + " is < 5");
                }
        }
}
```

Frequent Issues (I)

The signature of the *main* method *cannot* be modified.

```
public static void main(String[] arguments) {
                        ...
}
```

Frequent Issues (II)

Return values: if you declare that the method is not *void*, then it has to return something!

```
public static int pay(double basePay, int hours) {
    if (basePay < 8.0)    return -1;
    else if (hours > 60)  return -1;
    else {
        int salary = 0;
        ...
        return salary
    }
}
```

Frequent Issues (III)

Don't create duplicate variables with the same name

```
public static int pay(double basePay, int hours) {
   int salary = 0;      // OK
   …
   int salary = 0;      // salary already defined!!
   …
   double salary = 0; //salary already defined!!
   …
}
```

Assignment: FooCorporation

Main Method to print pay based on base pay and hours worked

Overtime: More than 40 hours, paid 1.5 times base pay

Minimum Wage: $8.00/hour

Maximum Work: 60 hours a week

Assignment: FooCorporation

Foo Corporation needs a program to calculate how much to pay their employees.

1. Pay = hours worked x base pay
2. Hours over 40 get paid 1.5 the base pay
3. The base pay must be no less than $8.00
4. The number of hours must be no more than 60

Questions from last lecture?

These slides are from:

- **6.092 Introduction to Programming in Java, January (IAP) 2010, MIT OpenCourseWare http://ocw.mit.edu**