



UCT Department of Computer Science  
Computer Science 1015F

# Java Basics



Hussein Suleman <hussein@cs.uct.ac.za>  
March 2007

## Java: the story so far ...

- A variable is a memory location where data can be stored. It usually has a name/identifier and data can be stored in it or retrieved from it.
  - e.g., `int i = 12;`
- Data types (`int`, `char`, etc.) specify what kinds of data can be manipulated.
- Arithmetic expressions are evaluated to numbers.
  - e.g., `j = i + 12`
- Boolean expressions are evaluated to true or false.
- Strings are sequences of characters.
- Input is done using the Scanner
- Output is done using System.out
- Instruction execution order can be controlled using selection (`if`, `if-else`, `switch`) and iteration (`for`, `while`, `do-while`)



## Software Testing

---

- Test Cases
- Tracing
- Assertions
- Debugging

## Test Cases

---

- Based on Input:
  - Choose one value in each **equivalence classes** – sets of input values that behave similarly.
  - Choose values on either side of and at **boundary values** – values between equivalence classes.
- Based on Code:
  - **Path coverage**: choose values that test every possible path through the statements at least once.
  - **Statement coverage**: Choose values that test every possible statement at least once.

## Test Cases Example

```
// Software Testing sample
// hussein suleman
// CSC1015F
// 19 March 2007

import java.util.Scanner;

class SoftwareTesting
{
    public static void main ( String [] args )
    {
        // get a number
        Scanner scan = new Scanner (System.in);
        int input = scan.nextInt();
        input += 10;

        // check if it is small or large
        if (input < 20)
            System.out.print ("small number");
        else
            System.out.print ("large number");

        // check if it is divisible by 5
        if (input % 5 == 0)
            System.out.println (" divisible by 5");
    }
}
```

### *equivalence classes:*

small multiple of 5: 5  
small non-multiples of 5: 3  
large multiple of 5: 25  
large non-multiples of 5: 23

### *boundary values:*

4, 5, 6, 9, 10, 11, 14, 15, 16

### *statement coverage:*

5, 14

### *path coverage:*

5, 7, 13, 20

## Assertions

- In Java a programmer can specify conditions that must always be satisfied at particular points (**invariants**) or the program produces an error. This is an **assertion**.

- Example:

```
assert (input > 0);
```

## Tracing

- Insert temporary statements into code to output values during calculation.
- Very useful when there is no debugger!

- Example:

```
int x = y*y*2;  
int z = x+5;
```

trace instruction

```
System.out.println (z);  
if (z == 13)  
{  
    ...  
}
```

## Debugging

- **Debugging** is the process of finding **errors** or **bugs** in the code.
- A **debugger** is a tool for executing an application where the programmer can carefully control execution and inspect data.
- Features include:
  - step through code one instruction at a time
  - viewing variables
  - insert and remove breakpoints to pause execution

## Problem

---

- Write a program to calculate the value of  $\sin(x)$  for any real value of  $x$ . Use the infinite Taylor series approximation:

$$\sin(x) \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!}.$$

- Extend your program to draw a  $\sin(x)$  graph using ASCII art.

## Solution

---

- First calculate  $\sin(x)$ .
  - Use variables for numerator, denominator, factorial, etc. and loop over terms until the term is very small.
- Add a loop over all  $x$  values to calculate a list of  $\sin(x)$  values.
- Add a loop over all  $y$ -axis values to visit each  $(x,y)$  pair and check if the graph intersects, i.e., if  $y = \sin(x)$  approximately
- Add axes if desired.