Elementary Programming
Part1 : Variables, Common Errors, Tracing

Course: CPSC 1150

Instructor: Dr. Bita Shadgar

Lecture 3

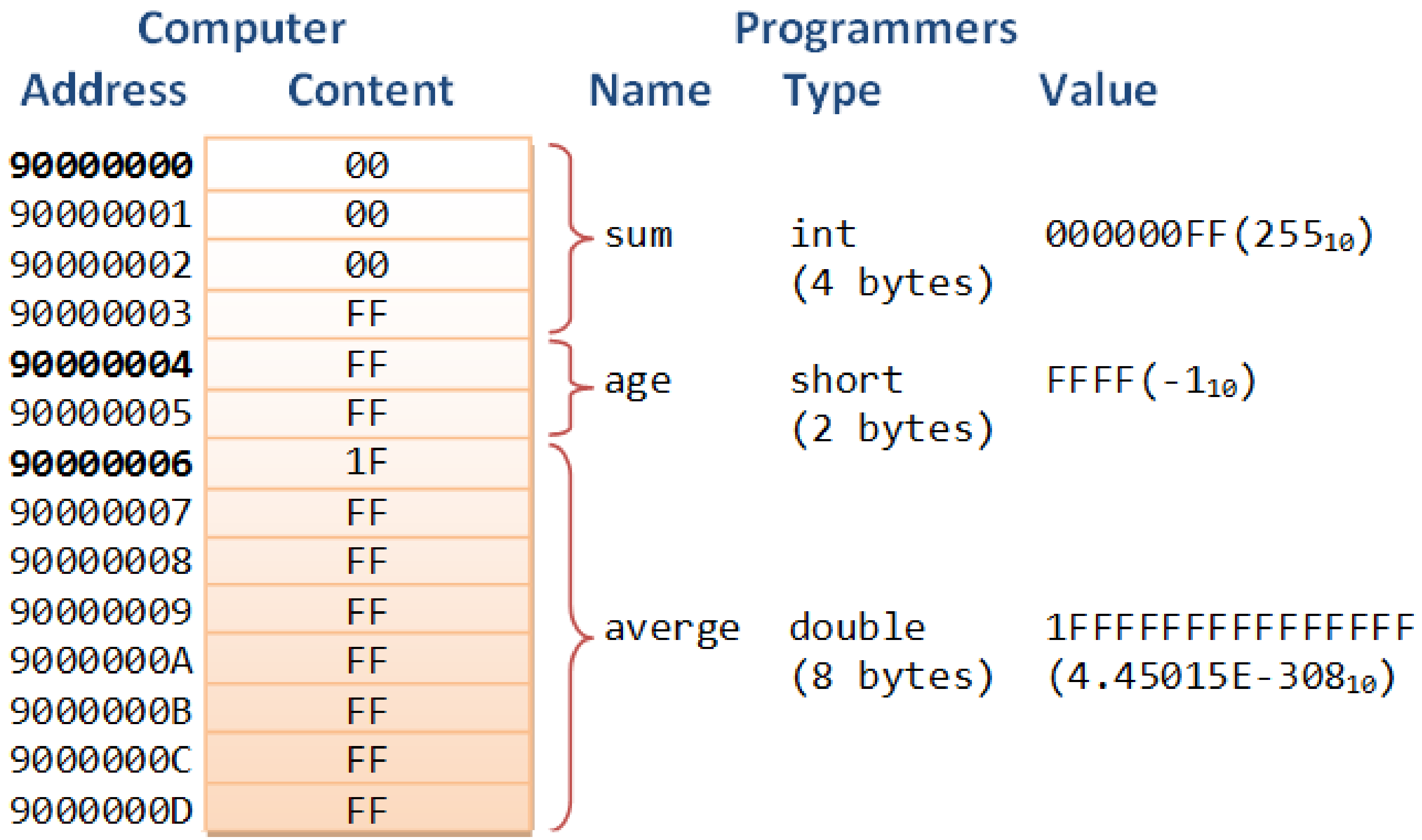
Learning Outcomes

- Name and define variables
- Declare a variable in Java
- Assign a value to a variable
- Identify different primitive data type in java and their domain
- Write assignment with expression
- Define named constant
- Recognize integer overflow as a common error
- Recognize roundoff error as a common error
- Trace program execution

A program that does something (better)!

- Question: How do we make programs that do anything more exciting than printing a pre-determined message to the console?
- Answer: By **storing and manipulating information**, using variables.

Storing Information in Memory



Computer		Programmers		
Address	**Content**	**Name**	**Type**	**Value**
90000000	00	sum	int (4 bytes)	000000FF(255_{10})
90000001	00			
90000002	00			
90000003	FF			
90000004	FF	age	short (2 bytes)	FFFF(-1_{10})
90000005	FF			
90000006	1F	averge	double (8 bytes)	1FFFFFFFFFFFFFFF ($4.45015E-308_{10}$)
90000007	FF			
90000008	FF			
90000009	FF			
9000000A	FF			
9000000B	FF			
9000000C	FF			
9000000D	FF			

Note: All numbers in hexadecimal

Variables

- Used to store data
- A variable has:
  - A name which represents its dress in memory
  - A data type
  - A fixed storage size
  - A value

Example

You could have a variable named `radius` of data type `float` (floating-point number) which has value `5.6`. The size of a `float` is 4 bytes (32 bits).

Identifiers

Definition

An **identifier** is a name for a variable (or object/class).

- Must consist of a sequence of letters, digits, underscores (_), and/or dollar signs ($)
- Cannot start with a digit
- Cannot be a Java keywords, or the words true, false, or null
- Case-sensitive, i.e., numStudents is not the same as numstudents

Naming conventions for variable identifiers

– Begin with a lowercase letter

– Be descriptive, but concise

– If an identifier has multiple words, capitalize the first letter of all but the first word (camel case)

Example

Some good variable names: `length, studentNum, netWeight, ageInDays, tempInCelsius`

Bad variable names

– Don’t make the names too long, or too hard to remember
– Avoid using two very similar identifiers in the same program

Example

Some bad variable names: fma03jga, Length, nineDigitStudentNumberOfProgramUser, aGeInDaYs

Data types

- A **type of item (data)** that can be stored
- How much **memory** an item occupies
- What **types of operations** can be performed on data

Keyword	Size	Description	Value
byte	1 byte	Byte-length integer	-128 to127
short	2 bytes	Short integer	-32768 to 32767
int	4 bytes	Integer	-2147483648 to 2147483647
long	8 bytes	Long integer	-2^{63} to $2^{63}-1$
float	4 bytes	Single-precision floating point	Negative value: -3.4028E+38 to -1.4E-45 Positive value: 1.4E-45 to 3.4028E+38
double	8 bytes	Double-precision floating point	Negative value: -1.7976E308 to -4.9E-324 Positive value: 4.9E-324 to 1.7976E308
char	2 bytes	A single character	All of the characters
boolean	Not-defined	A Boolean value	true or false

Declaring a variable

- A variable **declaration** “creates” and names a variable
- Tells CPU to allocate a certain amount of space in memory
- You **can not** assign a value to a variable without declaring it first
  - How much space depends on the data type

Sample declaration

```
int  k;
long studentNum;
float radius;
double netWeight;
```

Note: The above variables have no values yet.

Shortcuts for declarations

– You can declare multiple variables of the same type in one statement
– The name of the type only shows up once
– The variables must be separated by commas

Sample declarations

```
int i, j, k;
float radius, area;
```

Assigning a value to a variable

Syntax for assignment

```
variableName  =  value;
```

- An **assignment statement** gives a value to a variable
  - The variable must be declared first
  - The variable needs to be on the left of the equals sign
- The assigned value must be of the correct data type
  - For numeric types, there is some wiggle room here – we'll see this next week

Sample assignments

```
k = 0;
studentNum = 100271362;
radius  = 4.2;
netWeight = 0.00059288;
```

Misusing variables

Question:

- What do you think will happen if you attempt to use a variable that hasn’t been declared?
- What about a variable that has been declared but not initialized?

Assignments with expressions

– Variables can be assigned an entire **expression** involving other variables

```
area = 3.14159 * radius * radius;
// * is the  multiplication operator
```

– Variables can even be assigned in terms of themselves

– The right-hand side refers to the variable's previous value, before the assignment occurs

```
x = x + 1;
// adds 1 to x and stores this new value in x
```

More declarations and assignments

- If a variable already has a value, you can assign a new one
  - The old value will be lost
  - Do not re-declare the variable when re-assigning it
- Often it’s convenient to declare and **initialize** (assign a value for the first time) a variable all in one step

Example

```
float radius = 4.2; // Combined declaration and assignment
int i = 1, j = 3, k = -4; /* Several declarations, assignments
in one statement must be separated by commas */
```

How assignments work

- Assignment statements do two things:
  - Evaluate the expression to the right of the =
  - Store that value in the variable to the left of the =

  Tip:It might help to read assignment statements from right to left.
- Because of Item 1, it’s possible (and correct) to do things like:

```
System.out.println(radius = 4.2);
```

```
initialTemp = finalTemp = 21.7;
```

Question: How can we separate each of those statements into two statements?

Named constants

Syntax for declaring constants

```
final datatype CONSTANT_NAME= value;
```

- Sometimes you need a fixed constant to perform calculations
- Once the value is set, it cannot be changed (use final as modifier)
- Naming conventions for constants:
  - Use all uppercase letters
  - Multiple words should be separated by underscores i.e., MINUTES_PER_HOUR

Example

A program with circles might use pi (≈ 3.14159). At the beginning of the `main` method, you could declare a constant as follows:

```
final double PI = 3.14159;
```

Overflow Integer

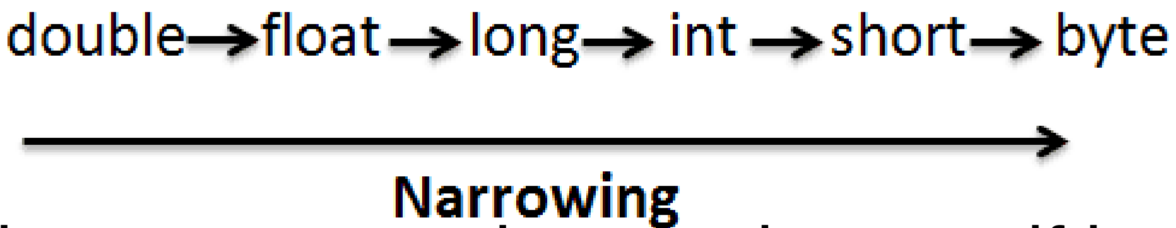
```
public class OverflowError {
   public static void main(String[] args){
      byte num = 120;
      System.out.println("The num is " + num);
      num = num + 10;
      System.out.println("The new num is " + num);
   }
}
```

```
$javac OverflowError.java
OverflowError.java:5: error: incompatible types: possible lossy conversion
from int to byte num = num + 10;
1 error
```

– Use explicit casting to convert a larger type to a smaller type.

Numeric Type Conversion (Casting)

- When you are assigning a **larger type** to a **smaller type**, then Explicit Casting is required



- Automatic type conversion can happen if both type are compatible and **targe type** is **larger** than **source type**.

Example

```
int theInt = 138;
byte aByte = (byte) theInt;  //aByte is -118, explicit casting
short  aShort = aByte;       //aShort is -118, implicit casting
```

Roundoff Error

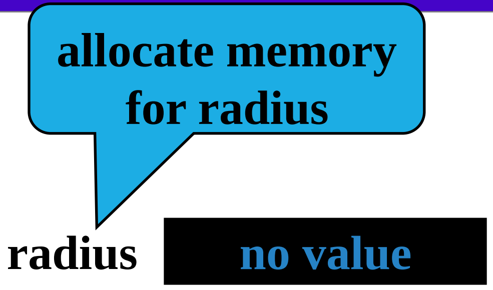
```java
public class RoundoffError{
  public static void main(String[] args) {
    System.out.print("1.0 - 0.1 - 0.1 - 0.1 - 0.1 - 0.1 = ");
    System.out.println(1.0 - 0.1 - 0.1 - 0.1 - 0.1 - 0.1);
    System.out.println("1.0 - 0.9 = " + (1.0 - 0.9));
  }
}
```

```
$javac RoundoffError.java
$java RoundoffError
1.0 - 0.1 - 0.1 - 0.1 - 0.1 - 0.1 = 0.5000000000000001
1.0 - 0.9 = 0.09999999999999998
```

– Calculation involving floating-point numbers are approximated, because these numbers are not stored with complete accuracy.

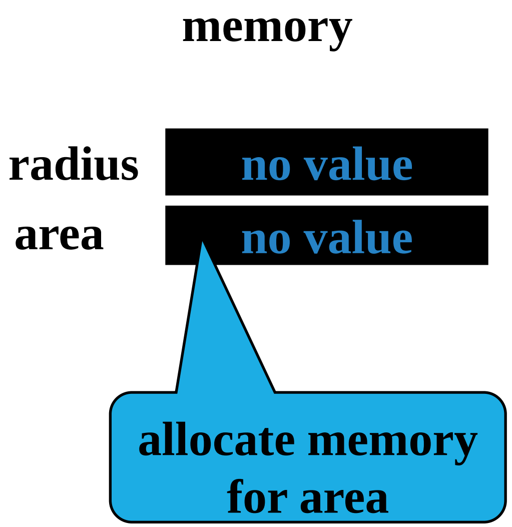
Tracing program execution

```
public class ComputeArea {
  /** Main method */
  public static void main(String[] args) {
    double radius;
    double area;
    // Assign a radius
    radius = 20;
    // Compute area
    area = radius * radius * 3.14159;
    // Display results
    System.out.println("The area for the circle of radius " +
                                radius + " is " + area);
  }
}
```



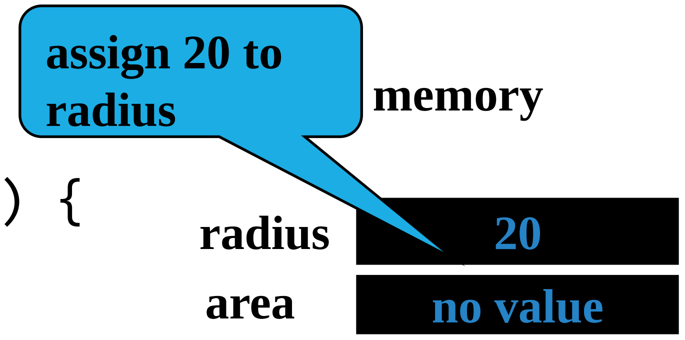
Tracing program execution

```
public class ComputeArea {
  /** Main method */
  public static void main(String[] args) {
    double radius;
    double area;
    // Assign a radius
    radius = 20;
    // Compute area
    area = radius * radius * 3.14159;
    // Display results
    System.out.println("The area for the circle of radius " +
                       radius + " is " + area);
  }
}
```



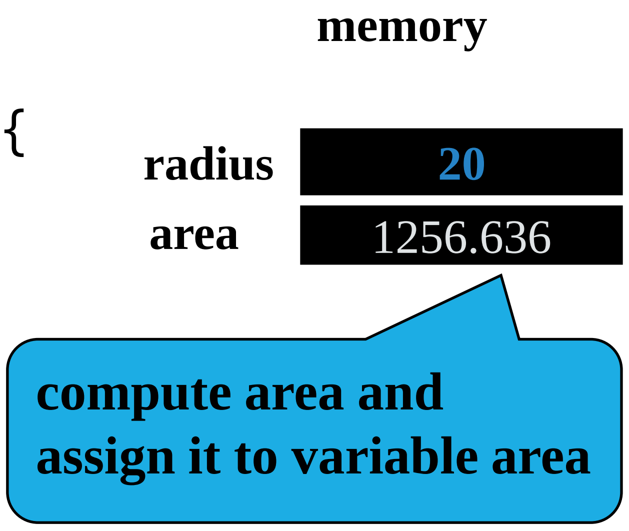
Tracing program execution

```
public class ComputeArea {
  /** Main method */
  public static void main(String[] args) {
    double radius;
    double area;
    // Assign a radius
    radius = 20;
    // Compute area
    area = radius * radius * 3.14159;
    // Display results
    System.out.println("The area for the circle of radius " +
                                radius + " is " + area);
  }
}
```



Tracing program execution

```
public class ComputeArea {
  /** Main method */
  public static void main(String[] args) {
    double radius;
    double area;
    // Assign a radius
    radius = 20;
    // Compute area
    area = radius * radius * 3.14159;
    // Display results
    System.out.println("The area for the circle of radius " +
                                radius + " is " + area);
  }
}
```



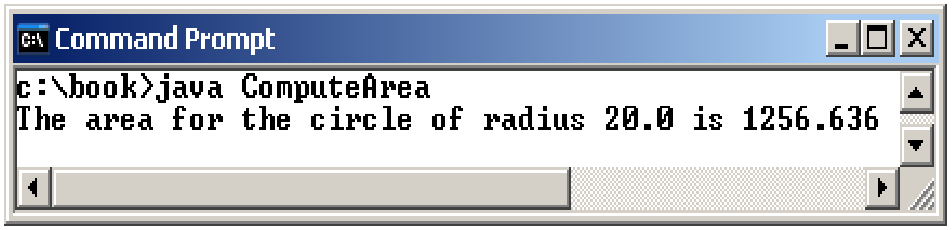
Tracing program execution

```
public class ComputeArea {
  /** Main method */
  public static void main(String[] args) {
    double radius;
    double area;
    // Assign a radius
    radius = 20;
    // Compute area
    area = radius * radius * 3.14159;
    // Display results
    System.out.println("The area for the circle of radius " +
                    radius + " is " + area);
  }
}
```

memory

radius	20
area	1256.636

print a message to the console



```
c:\book>java ComputeArea
The area for the circle of radius 20.0 is 1256.636
```

More Practice

- Declare a variable in one statement
- Declare more than one variable in one statement
- Assign a value a pre-declared variable
- Declare and initialize a float variable in one statement
- Declare and initialize more than one int variable in one statements
- Rewrite the following statement in three statements

  p= d = r = 10;

- Write a statement that assigns a float variable (eg. aFloat) to an integer variable (e.g. anInt)
- Define a named constant for the tax rate (12%)