CSY2030
Systems Design & Development
Revision of Java 1

Overview of Lecture

- Today we will revise the following in Java:
  - The *main* method
  - Variables
  - *for* loops
  - *while* loops
  - *do-while* loops
  - *if* statements
  - *switch* statements

The main method

- Code in the *main()* method is executed line by line
- The *main()* method is the only one ever called automatically

```
package csy2030;

public class CYS2030 {

public static void main(String[] args) {
		// TODO Auto-generated method stub
		System.out.println("One");
		System.out.println("Two");
	}

}
```

Java Variables

- Every variable needs a data type and a name
- Variables are declared using the syntax

  data-type name;
- They cannot be used until they're defined
- ***int myVariable;*** will declare a variable called *myVariable* that can be used to store integer values
- You only need to declare a variable once!

```
package csy2030;

public class CYS2030 {

public static void main(String[] args)
{
        int myVariable;
        myVariable = 3;

        System.out.println(myVariable)
;
}

}
```

Java Variables

- You can have shorthand variable declarations

```
package csy2030;

public class CYS2030 {

public static void
main(String[] args) {
        int myVariable;
        myVariable = 3;

        System.out.println(myVa
riable);
}

}
```

```
package csy2030;

public class CYS2030 {

public static void main(String[]
args) {
        int myVariable = 3;

        System.out.println(myVar
iable);
}

}
```

Java Variables

- Numerical variables can be used like algebra for mathematical operations:
  - + addition
  - - subtraction
  - / division
  - * multiplication

```
package csy2030;

public class CYS2030 {

public static void main(String[] args) {
        int myVariable;
        myVariable = 3;

        int myVariable2;
        myVariable2 = 4;

        System.out.println(myVariable +
        myVariable2);
}

}
```

Java String Variables

- Strings are a series of characters
- They're used for storing text in Java
- You can declare a string variable using

  String myVariable;
- Strings are defined in quotes

```
package csy2030;

public class CYS2030 {

public static void main(String[]
args) {
        String myVariable;
        myVariable = "some text";

        System.out.println(myVaria
ble);
}

}
```

Java String Variables

- Avoid the following (it won't compile):

```
package csy2030;

public class CYS2030 {

public static void main(String[] args)
{
    String myVariable;
         myVariable = "Bob said
"hello"";

    }

System.out.println(myVariable);
}
```

- Use \ instead:

```
package csy2030;

public class CYS2030 {

public static void main(String[] args)
{
     String myVariable;
          myVariable = "Bob said
\"hello\"";

   }

}
System.out.println(myVariable);
```

Java for loops

- Loops can be used to run the same code a number of times.
- This code will print *Hello* ten times
- Any code between the opening and closing brace will be run on each iteration

```
package csy2030;

public class CYS2030 {

public static void main(String[] args)
{
     for (int i = 0; i < 10; i++) {
        System.out.println("Hello");
     }

}
}
```

Java for loops

- It's possible to make use of the loop counter inside the loop
- The variable declared in the first part of the for statement (in this example called *i*) will store the number of the current iteration
- The counter must be an integer!

```
package csy2030;

public class CYS2030 {

public static void main(String[]
args) {
   for (int i = 0; i < 10; i++) {
        System.out.println(i);
    }
}

}
```

Java for loops

- A *for* loop has three parts.
- The first is the starting number

```
package csy2030;

public class CYS2030 {

public static void main(String[] args)
{
        for (int i = 0; i < 10; i++) {
                System.out.println(i);
        }
}

}
```

Java for loops

- The second is the condition
- The condition will be evaluated on each iteration and while it evaluates to true, the loop will continue
- This can be read as “while the counter is less than ten”

```
package csy2030;

public class CYS2030 {

public static void main(String[] args)
{
        for (int i = 0; i < 10; i++) {
                System.out.println(i);
        }
}

}
```

Java for loops

- The final part is the modifier, this will be executed at the end of each iteration
- ***i***++ means increment by one but this can be any mathematical expression.

```
package csy2030;

public class CYS2030 {

public static void main(String[]
args) {
        for (int i = 0; i < 10; i++) {
                System.out.println(i);
        }
}

}
```

Java while loops

- The while statement evaluates an *expression*,
- If the expression evaluates to true, the while statement executes the *statement*(s) in the while block.
- The while statement continues testing the expression and executing its block until the expression evaluates to false.

```
package csy2030;

public class CYS2030 {

public static void main(String[] args)
{
        int i = 0;
        while(i < 10) {
                System.out.println(i);
                i++;
        }
}

}
```

Java do-while loops

- *do-while* loops evaluates its expression at the bottom of the loop instead of the top.
- Therefore, the statements within the do block are always executed at least once

```
package csy2030;

public class CYS2030 {

public static void main(String[] args)
{
        int i = 0;

        do {
                System.out.println(i);
                i++;
        } while(i < 10);
}

}
```

Java if statements

- An *if* statement can be used to have the program inspect the value of a variable and execute some code if the condition evaluates to true
- Used with following operators:
  - == Equal to
  - != Not equal to
  - > greater than
  - >= greater than or equal
  - < less than
  - <= less than or equal
- This code will only print the number if it's equal to three

```
package csy2030;

public class CYS2030 {

public static void main(String[]
args) {
    for (int i = 0; i < 10; i++) {
        if (i == 3) {

    System.out.println("Three");
}

}
```

Java if statements

- An else statement can be added to an if statement
- This contains code that will be run when the condition is not met e.g

```
package csy2030;
public class CYS2030 {

    public static void main(String[] args) {
        for (int i = 0; i < 10; i++) {
        if (i == 3) {
                System.out.println("Three");
        else {
                System.out.println(“Not Three");
        }
    }
 }
```

Java if statements

- An else statement can be used to check a second condition and evaluate the value of the variable again e.g

```
package csy2030;
public class CYS2030 {
    public static void main(String[] args) {
        for (int i = 0; i < 10; i++) {
            if (i == 3) {
                System.out.println("Three");
            }
            else if (i == 4) {
                System.out.println(“Four");
            }
            else {
                System.out.println(“Not Three or Four");
                }
         }
    }
}
```

Java switch statements

- The *switch* statement is an alternative to using an if statement

```
package csy2030;

public class CYS2030 {

public static void main(String[] args) {
  int i;
  i = 3;
  if (i == 1) {
          System.out.println("One");
  else (i == 2) {
          System.out.println("Two");
  else (i == 3) {
          System.out.println("Three");
  else {
          System.out.println("Invlaid");
    }
  }
}
```

```
package csy2030;

public class CYS2030 {

public static void main(String[] args) {
  int i;
  i = 3;
  Switch (i){
     case 1: System.out.println("One");
          break;
      case 2:System.out.println("Two");
          break;
      case 3:
   System.out.println("Three");
          break;
     default:
   System.out.println("Invalid");
  }        break;
 } }
```