Digital Skills Academy

FUNDAMENTALS OF PROGRAMMING

LECTURE 9

http://www.projectcartoon.com/cartoon/2

Conor O'Reilly

This Session

6:00 – 7:00	LAB
7:00 – 8:00	Lecture – review assignment
8:00 – 8:15	Break
8:15 – 9:00	LAB
9:00 – 10:00	LAB

Mon 29-04-2013

April 1	May 2	May 3	May 4	May 5	June 6	June 7
29 Mon	**06 Mon**	**13 Mon**	**20 Mon**	**27 Mon**	**03 Mon**	**10 Mon**
9 - 12	9 - 12	9 - 12	9 - 12	9 - 12	9 - 12	9 - 12
12 - 2	12 - 2	12 - 2	12 - 2	12 - 2	12 - 2	12 - 2
2-4	2-4	2-4	2-4	2-4	2-4	2-4
4-6	4-6	4-6	4-6	4-6	4-6	4-6 mark eLABbooks
6-8	6-8	6-8	6-8	6-8	6-8	6-8 mark eLABbooks
8-10	8-10	8-10	8-10	8-10	8-10	8-10
30 Tue	**07 Tue**	**14 Tue**	**21 Tue**	**28 Tue**	**04 Tue**	**11 Tue**
9 - 12	9 - 12	9 - 12	9 - 12	9 - 12	9 - 12	9 - 12
12 - 2	12 - 2	12 - 2	12 - 2	12 - 2	12 - 2	12 - 2
2-4	2-4	2-4	2-4	2-4	2-4	2-4
4-6	4-6	4-6	4-6	4-6	4-6	4-6
6-8	6-8	6-8	6-8	6-8	6-8	6-8
8-10	8-10	8-10	8-10	8-10	8-10	8-10
01 Wed	**08 Wed**	**15 Wed**	**22 Wed**	**29 Wed**	**05 Wed**	**12 Wed**
9 - 12	9 - 12	9 - 12	9 - 12	9 - 12	9 - 12	9 - 12
12 - 2	12 - 2	12 - 2	12 - 2	12 - 2	12 - 2	12 - 2
2-4	2-4	2-4	2-4	2-4	2-4	2-4
4-6	4-6	4-6	4-6	4-6	4-6	4-6 mark eLABbooks
6-8	6-8	6-8	6-8	6-8	6-8	6-8 mark eLABbooks
8-10	8-10	8-10	8-10	8-10	8-10	8-10
02 Thu	**09 Thu**	**16 Thu**	**23 Thu**	**30 Thu**	**06 Thu**	**13 Thu**
9 - 12	9 - 12	9 - 12	9 - 12	9 - 12	9 - 12	9 - 12
12 - 2	12 - 2	12 - 2	12 - 2	12 - 2	12 - 2	12 - 2
2-4	2-4	2-4	2-4	2-4	2-4	2-4
4-6	4-6	4-6	4-6	4-6	4-6	4-6
6-8	6-8	6-8	6-8	6-8	6-8	6-8
8-10	8-10	8-10	8-10	8-10	8-10	8-10
03 Fri	**10 Fri**	**17 Fri**	**24 Fri**	**31 Fri**	**07 Fri**	**14 Fri**
9 - 12	9 - 12	9 - 12	9 - 12	9 - 12	9 - 12	9 - 12
12 - 2	12 - 2	12 - 2	12 - 2	12 - 2	12 - 2	12 - 2
2-4	2-4	2-4	2-4	2-4	2-4	2-4
4-6	4-6	4-6	4-6	4-6	4-6	4-6
6-8	6-8	6-8	6-8	6-8	6-8	6-8
8-10	8-10	8-10	8-10	8-10	8-10	8-10
					ELABOOK DUE 9th June 2013 11:59pm	

Here

Thinking in code **LAB1** **Scratch**	**First Program** **LAB2** **Basic Java** **No input just calc**	**Iterations & selections, Arrays** **LAB3 / 4** **Calculations** **Javabook** **Library**	**Classes and Objects** **LAB 5/6** **Instantiable classes**	**LABS** **Java Code Breaker Assignment**	**LABS**	**Java Assignment Self-directed** **eLABbooks assessed in class**

June	8	June	9	July	10	July	11	July	12	Reading Week July	Holidays July
17 Mon		**24 Mon**		**01 Mon**		**08 Mon**		**15 Mon**		**22 Mon**	**29 Mon**
9 - 12		9 - 12		9 - 12		9 - 12		9 - 12		9 - 12	9 - 12
12 - 2		12 - 2		12 - 2		12 - 2		12 - 2		12 - 2	12 - 2
2-4		2-4		2-4		2-4		2-4		2-4	2-4
4-6		4-6		4-6		4-6		4-6		4-6	4-6
6-8		6-8		6-8		6-8		6-8		6-8	6-8
8-10		8-10		8-10		8-10		8-10		8-10	8-10
18 Tue		**25 Tue**		**02 Tue**		**09 Tue**		**16 Tue**		**23 Tue**	**30 Tue**
9 - 12		9 - 12		9 - 12		9 - 12		9 - 12		9 - 12	9 - 12
12 - 2		12 - 2		12 - 2		12 - 2		12 - 2		12 - 2	12 - 2
2-4		2-4		2-4		2-4		2-4		2-4	2-4
4-6		4-6		4-6		4-6		4-6		4-6	4-6
6-8		6-8		6-8		6-8		6-8		6-8	6-8
8-10		8-10		8-10		8-10		8-10		8-10	8-10
19 Wed		**26 Wed**		**03 Wed**		**10 Wed**		**17 Wed**		**24 Wed**	**31 Wed**
9 - 12		9 - 12		9 - 12		9 - 12		9 - 12		9 - 12	9 - 12
12 - 2		12 - 2		12 - 2		12 - 2		12 - 2		12 - 2	12 - 2
2-4		2-4		2-4		2-4		2-4		2-4	2-4
4-6		4-6		4-6		4-6		4-6		4-6	4-6
6-8		6-8		6-8		6-8		6-8		6-8	6-8
8-10		8-10		8-10		8-10		8-10		8-10	8-10
20 Thu		**27 Thu**		**04 Thu**		**11 Thu**		**18 Thu**		**25 Thu**	**01 Thu**
9 - 12		9 - 12		9 - 12		9 - 12		9 - 12		9 - 12	9 - 12
12 - 2		12 - 2		12 - 2		12 - 2		12 - 2		12 - 2	12 - 2
2-4		2-4		2-4		2-4		2-4		2-4	2-4
4-6		4-6		4-6		4-6		4-6		4-6	4-6
6-8		6-8		6-8		6-8		6-8		6-8	6-8
8-10		8-10		8-10		8-10		8-10		8-10	8-10
21 Fri		**28 Fri**		**05 Fri**		**12 Fri**		**19 Fri**		**26 Fri**	**02 Fri**
9 - 12		9 - 12		9 - 12		9 - 12		9 - 12		9 - 12	9 - 12
12 - 2		12 - 2		12 - 2		12 - 2		12 - 2		12 - 2	12 - 2
2-4		2-4		2-4		2-4		2-4		2-4	2-4
4-6		4-6		4-6		4-6		4-6		4-6	4-6
6-8		6-8		6-8		6-8		6-8		6-8	6-8
8-10		8-10		8-10		8-10		8-10		8-10	8-10
						Java Program Due 14th July 2013 11:59pm				UCDWeb Prototype due 28th July 2013	

Intro to Web HTML/CSS **javascript** **bootstrap** **JQuery** **LAB** **-** **-**

Fundamentals of Programming

LAST SESSION

In groups of no more than 4, discuss

- **What topics did we cover?**
- **What did you learn?**
- **What stood out for you?**
- **Any issues?**

The spokesperson for the group will be the persons who's birthday is closest to the **20th of October** and who hasn't spoken before

Conor O'Reilly

char and Strings

```
char aCharacter = 'X';
```

Use single quotes

```
String name = "Sumatra";
```

name.charAt(3)

*is **a** as index starts at 0*

```
String aName = "Conor";
aName.length();
```

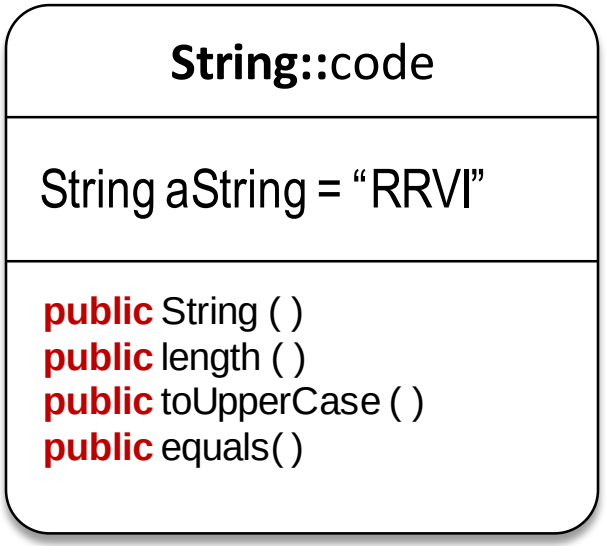
5

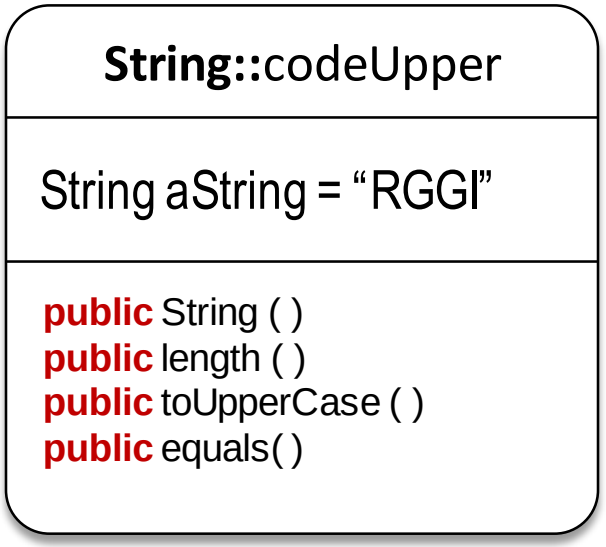
```
String aName;
aName.length();
```

NullPointerException

Convert to uppercase and see if equals

```
String code = "RRVI"
String codeUpper;
codeUpper = code.toUpperCase();
if(code.equals("RGGI"))
{
   System.out.println ("You Win");
}
else
{
   System.out.println ("You loose");
}
```





StringBuffer

- **A String object is immutable**, which means that once a String object is created, we cannot change it.
- We can read individual characters in a string, but we cannot add, delete, or modify characters of a String object.
  - Remember that the methods of the String class, such as toUpperCase and substring, do not modify the original string**; they return a new string**.
- Creating a new string from the old one will work for most cases, but sometimes manipulating the content of a string directly is more convenient.
  - Manipulation here means operations such as replacing a character, appending a string with another string, deleting a portion of a string, and so forth.

Sample StringBuffer Program

Replace all vowels in the sentence with 'X'.

```
char         letter;
String       inSentence          = inputBox.getString("Enter a sentence:");
StringBuffer tempStringBuffer    = new StringBuffer(inSentence);
int          numberOfCharacters = tempStringBuffer.length();

for (int index = 0; index < numberOfCharacters; index++){
     letter = tempStringBuffer.charAt(index);
     if (letter == 'a' || letter == 'A' || letter == 'e' || letter == 'E' ||
          letter == 'i' || letter == 'I' || letter == 'o' || letter == 'O' ||
          letter == 'u' || letter == 'U'  ){
          tempStringBuffer.setCharAt(index,'X');
     }
}
messageBox.show( tempStringBuffer );
```

Overview

Example Code

- Guess a number between 1 and 6
- Guess a number between 1 and 10
- Guess a vowel (a letter)
- Guess the planet (a word)

Java

GUESS A NUMBER 1 - 6

http://www.projectcartoon.com/cartoon/2

Conor O'Reilly

Approach

- Draw up a flowchart for the application
- Write the code in steps

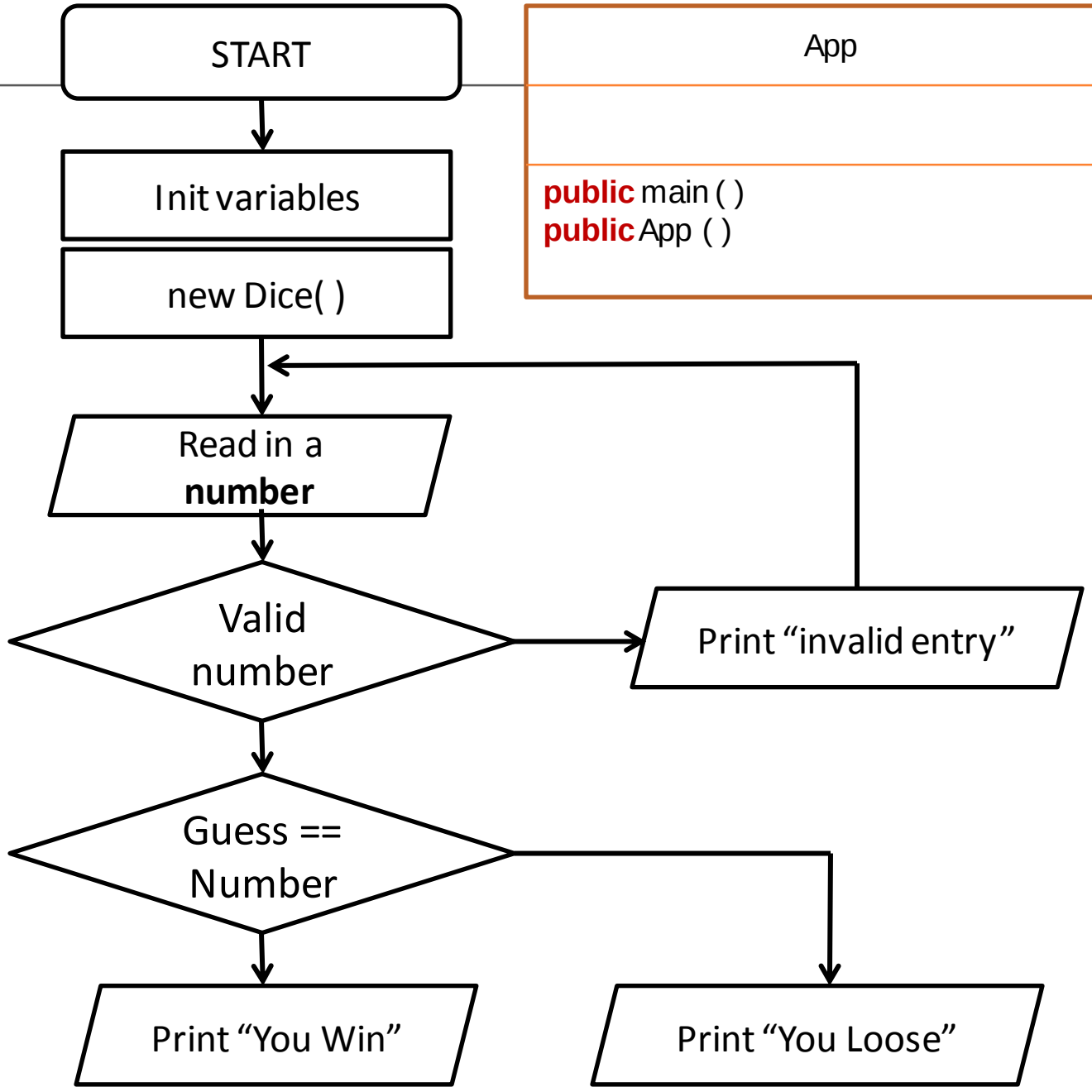
```
21_DiceRoll
22_DiceRoll4Times
23_DiceRollCheckSum
24_GuessANumberBetween1and6
25_GuessANumberBetween1and10
26_GuessANumberBetween1and10withLives
27_GuessANumberBetween1and10withLivesAndPlayMethod
28_GuessANumberBetween1and10withLivesAndPlayRetry
29_GuessANumberBetween1and10withLivesAndPlayRetrySeeNums
30_GuessANumberBetween1and10withLivesAndPlayRetrySeeNumsUsedBefore
31_GuessANumberBetween1and10withLivesAndPlayRetrySeeNumsUsedBeforeCheckInput
32_GuessALetter
```

Dice Class

Dice
private final int NUMBER_OF_SIDES **private** int faceValue
public Dice( ) **public** throwDice( ) **public** double getFaceValue( )

```
class Dice
{
    // DATA
    //------------------------------------------------------------
    //Private Constants
    final int NUMBER_OF_SIDES = 6;

    //Private Variables
    private int faceValue;


    // CONSTRUCTORS
    //------------------------------------------------------------
    public Dice()
    {
        this.faceValue = 0;      //zero if not thrown
    }

    // METHODS - behaviours
    //------------------------------------------------------------
    public void throwDice()
    {
        this.faceValue = 1 + (int) (Math.random() * NUMBER_OF_SIDES);
    }


    // METHODS - gets (accessors)  and sets (mutators)
    //------------------------------------------------------------
    public int getFaceValue()
    {
            return(this.faceValue);
    }

}
```

App



Get random number, ask for guess

```java
//create objects
theDice = new Dice();
someInput = new Scanner(System.in);

//get a random number for the user to guess
theDice.throwDice();
numberToGuess = theDice.getFaceValue();

//input: game intro
System.out.println("This is a number guessing game.");
System.out.println("Guess a number between 1 and 6.");

//input: enter a guess number
do
{
    System.out.print("Please enter a guess : ");
    theUsersInput = someInput.nextLine();

    try
    {
        //validate the input, convert from String to int
        integerEntered = Integer.parseInt(theUsersInput);
        invalidInput = false;
    }
    catch (Exception e)
    {
        System.out.println(" You entered " + theUsersInput + ", this is not a valid entry, retry. \n");
        invalidInput = true;
    }
}
while(invalidInput);
```

```java
//declare variable
int integerEntered;
boolean invalidInput;
int numberToGuess;
int numberOfLives;
int numbersEnteredArray[];    //to store previ

//declare objects
Scanner someInput;
String theUsersInput;
Dice theDice;

//initialise variables
integerEntered = 0;
invalidInput = true;
numberOfLives = 3;
numbersEnteredArray = new int[numberOfLives];
```

Loop until a valid number is entered

Convert String to an int

Guess correct or not

```
//processing : compare numbers
if( numberToGuess == integerEntered )
{
    System.out.println("\n YOU WIN - Good Guess !! the number was : " + numberToGuess);
}
else
{
    System.out.println("\n YOU LOOSE, the number was : " + numberToGuess);
}

//pause before exit
System.out.println(" \n Press enter to exit the program");
someInput.nextLine();

//close the program without error
System.exit(0);
```

Java

GUESS A NUMBER 1 – 10

NumberGenerator CLASS

http://www.projectcartoon.com/cartoon/2

Conor O'Reilly

NumberGenerator( )

```
class NumberGenerator
{
    // DATA
    //.....................................................................


    // CONSTRUCTORS
    //.....................................................................
    public NumberGenerator()
    {

    }

    // METHODS
    //.....................................................................
    public int getNumber()
    {
        return( 1 + (int) (Math.random() * 10) );
    }

}
```

```
//create objects
theNumberGenerator = new NumberGenerator();
someInput = new Scanner(System.in);

//get a random number for the user to guess
numberToGuess = theNumberGenerator.getNumber();

//input: game intro
System.out.println("This is a number guessing game.");
System.out.println("Guess a number between 1 and 10.");

//input: enter a guess number
do
{
    System.out.print("Please enter a guess : ");
    theUsersInput = someInput.nextLine();

    try
    {
        //validate the input, convert from String to int
        integerEntered = Integer.parseInt(theUsersInput);
        invalidInput = false;
    }
    catch (Exception e)
    {
        System.out.println(" You entered " + theUsersInput + ", this is not a valid entry, retry. \n");
        invalidInput = true;
    }
}
while(invalidInput);
```

NumberGenerator replacing Dice

Java

NUMBER OF LIVES

http://www.projectcartoon.com/cartoon/2

```
//loop the number of lives
livesLeft = numberOfLives;

for (int i = 0; i < numberOfLives; i=i+1)
{

    //input: enter a guess number
    do
    {
        System.out.print("\n  Please enter a guess : ");
        theUsersInput = someInput.nextLine();

        try
        {
            //validate the input, convert from String to int
            integerEntered = Integer.parseInt(theUsersInput);
            invalidInput = false;
        }
        catch (Exception e)
        {
            System.out.println(" You entered " + theUsersInput + ", this is not a valid entry, retry. \n");
            invalidInput = true;
        }
    }
    while(invalidInput);

    //processing : compare numbers
    if( numberToGuess == integerEntered )
    {
        guessed = true;
        break;
    }
    else
    {
        livesLeft = livesLeft - 1;
        System.out.println("\n Try again, guesses left: " + livesLeft);
        guessed = false;
    }
}
```

Number of lives loop

```
            //processing : compare numbers
            if( numberToGuess == integerEntered )
            {
                guessed = true;
                break;
            }
            else
            {
                livesLeft = livesLeft - 1;
                System.out.println("\n Try again, guesses left: " + livesLeft);
                guessed = false;
            }
        }

        //check if guessed or not
        if( guessed )
        {
            System.out.println("\n YOU WIN - Good Guess !! the number was : " + numberToGuess);
        }
        else
        {
            System.out.println("\n YOU LOOSE, the number was : " + numberToGuess);
        }

        //pause before exit
        System.out.println(" \n Press enter to exit the program");
        someInput.nextLine();

        //close the program without error
        System.exit(0);

    }//EOM-App()

}//EOC
```

Jump out of the loop with break

Press a key Then goes to exit

Java

PREPARE FOR PLAY AGAIN LOOP

http://www.projectcartoon.com/cartoon/2

Conor O'Reilly

```java
//.....................................................
public App()
{
    //initialise variables
    this.numberOfLives = 3;
    this.numbersEnteredArray = new int[this.numberOfLives];   //array = max number of lives
    this.livesLeft = 0;

    //create objects
    this.theNumberGenerator = new NumberGenerator();
    this.someInput = new Scanner(System.in);

    //playGame
    playGame();

    //pause before exit
    System.out.println(" \n Press enter to exit the program");
    someInput.nextLine();

    //close the program without error
    System.exit(0);
}

// METHODS
//.....................................................
private void playGame()
{
    this.guessed = false;
    this.integerEntered = 0;
    this.invalidInput = true;

    //get a random number for the user to guess
    numberToGuess = theNumberGenerator.getNumber();

    //input: game intro
    System.out.println("This is a number guessing game.");
    System.out.println("Guess a number between 1 and 10.");

    //loop the number of lives
    livesLeft = numberOfLives;

    for (int i = 0; i < numberOfLives; i=i+1)
    {
```

App
public main ( ) **public** App ( ) **private** playGame ( )

Move play into its own method

Java

APP( )
PLAYBOARD( )
PLAYGAME( )

http://www.projectcartoon.com/cartoon/2

App( )

```
// CONSTRUCTORS
//.....................................................
public App()
{
    //initialise variables
    this.numberOfLives = 3;
    this.numbersEnteredArray = new int[this.numberOfLives];   //array = max number of lives
    this.livesLeft = 0;
    this.tryAgain = false;

    //create objects
    this.theNumberGenerator = new NumberGenerator();
    this.someInput = new Scanner(System.in);

    //playGame
    playBoard();

    //pause before exit
    System.out.println(" \n Press enter to exit the program");
    this.someInput.nextLine();

    //close the program without error
    System.exit(0);
}
```

```
// METHODS
//.....................................................

/*=================================================
 * ask if want to play again once played once
 *
 *=================================================*/
private void playBoard()
{
    do
    {

        playGame();

        this.tryAgain = false;
        System.out.print("\n Play again (Y/N): ");
        this.theUsersInput = this.someInput.nextLine();

        // we get a String in, we only want the first character
        // a String is like an array, the first position starts at 0
        // theLetterIn is of type char

        this.theLetterIn = this.theUsersInput.charAt(0);

        //now comparing two characters

        if( (this.theLetterIn == 'Y') || ( this.theLetterIn == 'y') )
        {
            this.tryAgain = true;
        }

    }
    while(this.tryAgain);

}//EOM-playBoard()
```

App
public main ( ) **public** App ( ) **private** playBoard ( ) **private** playGame ( )

Put loop around method playGame() cleaner code

Check for Y or y

```java
/*================================================
 * Play the game
 *
 *================================================*/
private void playGame()
{
    this.guessed = false;
    this.integerEntered = 0;
    this.invalidInput = true;

    //get a random number for the user to guess
    this.numberToGuess = this.theNumberGenerator.getNumber();

    //input: game intro
    System.out.println("This is a number guessing game.");
    System.out.println("Guess a number between 1 and 10.");

    //loop the number of lives
    this.livesLeft = this.numberOfLives;

    for (int i = 0; i < this.numberOfLives; i=i+1)
    {

        //input: enter a guess number
        do
        {
            System.out.print("\n  Please enter a guess : ");
            this.theUsersInput = this.someInput.nextLine();

            try
            {
                //validate the input, convert from String to int
                this.integerEntered = Integer.parseInt(this.theUsersInput);
                this.invalidInput = false;
            }
            catch (Exception e)
            {
                System.out.println(" You entered " + this.theUsersInput + ", this is not a valid entry, retry. \n");
                this.invalidInput = true;
            }
        }
        while(this.invalidInput);
```

Nothing changed here

Java

HISTORY OF GUESSES ENTERED SO CAN'T GUESS SAME NUMBER TWICE

http://www.projectcartoon.com/cartoon/2

```
private void playGame()
{
    this.guessed = false;
    this.integerEntered = 0;
    this.invalidInput = true;

    int count = 0;

    //get a random number for the user to guess
    this.numberToGuess = this.theNumberGenerator.getNumber();

    //input: game intro
    System.out.println("\n---------------------------------");
    System.out.println("This is a number guessing game.");
    System.out.println("Guess a number between 1 and 10.");

    //set number of lives
    this.livesLeft = this.numberOfLives;

    //clear out input history array
    for (int i = 0; i < this.numberOfLives; i=i+1)
    {
        this.numbersEnteredArray[i] = 0;
    }

    //loop the number of lives
    for (int i = 0; i < this.numberOfLives; i=i+1)
    {
```

```
    //processing : compare numbers
    if( this.numberToGuess == this.integerEntered )
    {
        this.guessed = true;
        break;
    }
    else
    {
        this.livesLeft = this.livesLeft - 1;

        //put number entered into history array
        this.numbersEnteredArray[count] = this.integerEntered;
        count = count + 1;

        //show history of numbers entered
        System.out.print("\n Entered so far: ");

        //use j variable as within a loop that is using i
        for (int j = 0; j < count; j=j+1)
        {
            System.out.print(this.numbersEnteredArray[j] + "  ");
        }

        System.out.println("\n\n Try again, guesses left: " + this.livesLeft);
        this.guessed = false;
    }
```

First just store the numbers input

```
System.out.print("\n  Please enter a guess : ");
this.theUsersInput = this.someInput.nextLine();

try
{
    //validate the input, convert from String to int
    this.integerEntered = Integer.parseInt(this.theUsersInput);
    this.invalidInput = false;
}
catch (Exception e)
{
    System.out.println(" You entered " + this.theUsersInput + ", this is not a valid entry, retry. \n");
    this.invalidInput = true;
}

//check if entered that number before
for (int k = 0; k < this.numberOfLives; k=k+1)
{
    if ( this.numbersEnteredArray[k] == this.integerEntered )
    {
        System.out.println(" You entered the number : " + this.theUsersInput + " before, please pick a different number. \n");
        this.invalidInput = true;
    }
}
```

Loop through see if number input before

Java

ONLY ALLOW NUMBERS BETWEEN 1 AND 10 TO BE ENTERED

http://www.projectcartoon.com/cartoon/2

```
for (int i = 0; i < this.numberOfLives; i=i+1)
{

    //input: enter a guess number
    do
    {
        System.out.print("\n  Please enter a guess : ");
        this.theUsersInput = this.someInput.nextLine();

        try
        {
            //validate the input, convert from String to int
            this.integerEntered = Integer.parseInt(this.theUsersInput);
            this.invalidInput = false;
        }
        catch (Exception e)
        {
            System.out.println(" You entered " + this.theUsersInput + ", this is not a valid entry, retry. \n");
            this.invalidInput = true;
        }


        //check if number entered is within the allowed range
        if( (this.integerEntered < 1) || ( this.integerEntered > 10) )
        {
            System.out.println(" You entered " + this.theUsersInput + ", only 1 to 10 is allowed, please reenter. \n");
            this.invalidInput = true;
        }

        //check if entered that number before
        for (int k = 0; k < this.numberOfLives; k=k+1)
        {
            if ( this.numbersEnteredArray[k] == this.integerEntered )
            {
                System.out.println(" You entered the number : " + this.theUsersInput + " before, please pick a different r
                this.invalidInput = true;
            }
        }


    }
    while(this.invalidInput);
```

Between 1 to 10

Java

NOW GUESS A VOWEL

http://www.projectcartoon.com/cartoon/2

```java
class LetterGenerator
{
    // DATA
    //.....................................................

    private char [] allowedLetters = {'A','E','I','O','U'};


    // CONSTRUCTORS
    //.....................................................
    public LetterGenerator()
    {
        //no init to do in this constructor
    }

    // METHODS
    //.....................................................
    public char getLetter()
    {
        return( this.allowedLetters[ (int) (Math.random() * 5) ] );
    }


    public boolean checkLetter( char letterToCheck )
    {

        int matchCount = 0;

        for (int i = 0; i < allowedLetters.length; i=i+1)
        {
            if ( letterToCheck == allowedLetters[i] )
            {
                //should equal at least one
                matchCount = matchCount + 1;
            }
        }

        if ( matchCount == 0 )
        {
            //not in allowed letter set
            return false;
        }
        else
        {
            //in allowed letter set
            return true;
        }
    }
```

Checking the valid letters allowed is here in this class As this class knows what is allowed

```java
public static void main(String args[])
{
        App anApp = new App();
}


// DATA
//.....................................................................
private char guessLetter;
private char letterToGuess;
private boolean invalidInput;
private boolean guessed;
private boolean tryAgain;
private int numberToGuess;
private int numberOfLives;
private char lettersEnteredArray[];    //to store previous guess numbers entered
private int livesLeft;
private char theLetterIn;

//declare objects
private Scanner someInput;
private String theUsersInput;
private LetterGenerator theLetterGenerator;


// CONSTRUCTORS
//.....................................................................
public App()
{
    //initialise variables
    this.numberOfLives = 5;
    this.lettersEnteredArray = new char[this.numberOfLives];   //array = max number of lives
    this.livesLeft = 0;
    this.tryAgain = false;

    //create objects
    this.theLetterGenerator = new LetterGenerator();
    this.someInput = new Scanner(System.in);

    //playGame
    playBoard();

    //pause before exit
    System.out.println(" \n Press enter to exit the program");
    this.someInput.nextLine();

    //close the program without error
    System.exit(0);
}
```

char

New class letter generator

History now a char array

```
private void playGame()
{
    char initialLetterIn;
    this.guessed = false;
    this.invalidInput = true;

    int count = 0;

    //get a random number for the user to guess
    this.letterToGuess = this.theLetterGenerator.getLetter();

    //input: game intro
    System.out.println("\n-----------------------------------");
    System.out.println("This is a vowel guessing game.");
    System.out.println("Guess the vowel is it A,E,I,O or U.");

    //set number of lives
    this.livesLeft = this.numberOfLives;

    //clear out input history array
    for (int i = 0; i < this.numberOfLives; i=i+1)
    {
        // note use of single quotes not double quotes around characters
        // double quotes mean a String so "_" is a single letter String object
        // whilst '_' is a primitive character variable
        this.lettersEnteredArray[i] = '_';
    }
```

char

```
//loop the number of lives
for (int i = 0; i < this.numberOfLives; i=i+1)
{

    //input: enter a guess letter
    do
    {
        System.out.print("\n  Please enter a guess : ");
        this.theUsersInput = this.someInput.nextLine();

        //take the first character out of any string entered
        initialLetterIn = this.theUsersInput.charAt(0);

        //convert character to uppercase before start compars
        this.guessLetter = Character.toUpperCase(initialLetterIn);

        //check if the input was an allowed letter
        if( this.theLetterGenerator.checkLetter(this.guessLetter) )
        {
            this.invalidInput = false;
        }
        else
        {
            System.out.println(" You entered : " + this.theUsersInput.charAt(0) + " this is not a valid entry, try again. \n");
            this.invalidInput = true;

        }

        //TODO

        //check if entered that letter before
        for (int k = 0; k < this.numberOfLives; k=k+1)
        {
            if ( this.lettersEnteredArray[k] == this.guessLetter )
            {
                System.out.println(" You entered the letter : " + this.theUsersInput.charAt(0) + " before, please pick a different letter. \n");
                this.invalidInput = true;
            }
        }

    }
    while(this.invalidInput);
```

Ask the LetterGenerator object if the input is valid

```
            //processing : compare numbers
            if( this.letterToGuess == this.guessLetter )
            {
                this.guessed = true;
                break;
            }
            else
            {
                this.livesLeft = this.livesLeft - 1;

                //put letter entered into history array
                this.lettersEnteredArray[count] = this.guessLetter;
                count = count + 1;

                //show history of letters entered
                System.out.print("\n Entered so far: ");

                //use j variable as within a loop that is using i
                for (int j = 0; j < count; j=j+1)
                {
                    System.out.print(this.lettersEnteredArray[j] + "  ");
                }

                System.out.println("\n\n Try again, guesses left: " + this.livesLeft);
                this.guessed = false;
            }
        }

        //check if guessed or not
        if( this.guessed )
        {
            System.out.println("\n YOU WIN - Good Guess !! the letter was : " + this.letterToGuess);
        }
        else
        {
            System.out.println("\n YOU LOOSE, the letter was : " + this.letterToGuess);
        }

    }//EOM-play()

}//EOC
```

Check if letters used in a guess before

Java

FIND A PLANET (A WORD)

http://www.projectcartoon.com/cartoon/2

Conor O'Reilly

WordGenerator

```
class WordGenerator
{
    // DATA
    //.....................................................................

    private String [] wordList = { "mercuary","venus","earth","mars","jupiter","saturn","uranus","neptune","pluto"};


    // CONSTRUCTORS
    //.....................................................................
    public WordGenerator()
    {
        //no init to do in this constructor
    }

    // METHODS
    //.....................................................................
    public String getWord()
    {
        return( this.wordList[ (int) (Math.random() * 9) ] );
    }


}
```

```java
// DATA
//.....................................................................
private char guessLetter;
private String wordToGuess;
private StringBuffer guessedSoFar;
private boolean invalidInput;
private boolean guessed;
private int guessCount;
private int guessCountTotal;
private boolean tryAgain;
private int numberToGuess;
private int numberOfLives;
private char lettersEnteredArray[];    //to store previous guess numbers entered
private int livesLeft;
private char theLetterIn;

//declare objects
private Scanner someInput;
private String theUsersInput;
private WordGenerator theWordGenerator;


// CONSTRUCTORS
//.....................................................................
public App()
{
    //initialise variables
    this.numberOfLives = 5;
    this.lettersEnteredArray = new char[26];   //array = max number of letters in alphabeth
    this.livesLeft = 0;
    this.tryAgain = false;
    this.guessCount = 0;

    //create objects
    this.theWordGenerator = new WordGenerator();
    this.someInput = new Scanner(System.in);

    //playGame
    playBoard();

    //pause before exit
    System.out.println(" \n Press enter to exit the program");
    this.someInput.nextLine();

    //close the program without error
    System.exit(0);
}
```

StringBuffer : show letters guessed in word

```
private void playGame()
{
    char initialLetterIn;
    this.guessed = false;
    this.invalidInput = true;
    this.guessCountTotal = 0;

    int count = 0;

    //get a random number for the user to guess
    this.wordToGuess = this.theWordGenerator.getWord();

    //init guessed so far with blank characters
    this.guessedSoFar = new StringBuffer();
    for (int i = 0; i < this.wordToGuess.length(); i=i+1)
    {
        guessedSoFar.append("_");
    }

    //input: game intro
    System.out.println("\n-----------------------------------");
    System.out.println("This is a planet guessing game.");
    System.out.println("But you must guess the planet a letter at a time");
    System.out.println("Hint: " + putInSpaces(this.guessedSoFar.toString()));

    //set number of lives
    this.livesLeft = this.numberOfLives;

    //clear out input history array
    for (int i = 0; i < 26; i=i+1)
    {
        // note use of single quotes not double quotes around characters
        // double quotes mean a String so "_" is a single letter String object
        // whilst '_' is a primitive character variable
        this.lettersEnteredArray[i] = '_';
    }
```

Show how much of the word has been guessed

```
_____________________________________
This is a planet guessing game.
But you must guess the planet a letter at a time
Hint: _ _ _ _ _ _ _

  Please enter a guess :
```

```java
//loop the number of lives
while( this.livesLeft > 0 )
{

    //input: enter a guess letter
    do
    {
        this.invalidInput = false;
        System.out.print("\n  Please enter a guess : ");
        this.theUsersInput = this.someInput.nextLine();

        //take the first character out of any string entered
        initialLetterIn = this.theUsersInput.charAt(0);

        //convert character to lowercase before start compars
        this.guessLetter = Character.toLowerCase(initialLetterIn);

        //check if entered that letter before
        for (int k = 0; k < 26; k=k+1)
        {
            if ( this.lettersEnteredArray[k] == this.guessLetter )
            {
                System.out.println(" You entered the letter : " + this.theUsersInput.charAt(0) + " before, please pick a different
                this.invalidInput = true;
            }
        }

    }
    while(this.invalidInput);

    //loop through each character of the word see if we have one or more matches
    this.guessCount = 0;
    for (int w = 0; w < this.wordToGuess.length(); w=w+1)
    {
        if( this.wordToGuess.charAt(w) == this.guessLetter)
        {
            this.guessCount = this.guessCount + 1;
            this.guessedSoFar.setCharAt(w, this.guessLetter);
        }

    }
```

Get input, check character not previously input

Count the number of times a character match is found in a word

```
    //if get any letters correct, tell or loose a life
    if( this.guessCount > 0)
    {
        System.out.println("\n Hint: " + putInSpaces(this.guessedSoFar.toString()));
        System.out.println("\n\n good guess, guesses left: " + this.livesLeft);
        this.guessCountTotal = this.guessCountTotal + this.guessCount;
    }
    else
    {
        this.livesLeft = this.livesLeft - 1;

        //put letter entered into history array
        this.lettersEnteredArray[count] = this.guessLetter;
        count = count + 1;

        //show history of letters entered
        System.out.print("\n Entered so far: ");

        //use j variable as within a loop that is using i
        for (int j = 0; j < count; j=j+1)
        {
            System.out.print(this.lettersEnteredArray[j] + "  ");
        }

        System.out.println("\n Hint: " + putInSpaces(this.guessedSoFar.toString()));
        System.out.println("\n\n Try again, guesses left: " + this.livesLeft);
    }

    //whole word guessed?
    //number of characters guseesd = length of word
    if( this.guessCountTotal == this.wordToGuess.length())
    {
        this.guessed = true;
        //jump out of lives loop as all letters guessed
        break;
    }

} //for lives loop

//check if guessed or not after run out of lives
if( this.guessed )
{
    System.out.println("Your Guess: " + putInSpaces(this.guessedSoFar.toString()));
    System.out.println("\n YOU WIN - Good Guess !! the word was : " + this.wordToGuess);
}
else
{
    System.out.println("Your Guess: " + putInSpaces(this.guessedSoFar.toString()));
```

If a letter guessed, show

Loose a life

```
-----------------------------------
This is a planet guessing game.
But you must guess the planet a letter at a time
Hint: _ _ _ _ _

 Please enter a guess : e

Hint: _ e _ _ _

good guess, guesses left: 5

 Please enter a guess : v

Hint: v e _ _ _

good guess, guesses left: 5

 Please enter a guess : n

Hint: v e n _ _

good guess, guesses left: 5

 Please enter a guess : u

Hint: v e n u _

good guess, guesses left: 5

 Please enter a guess : s

Hint: v e n u s

good guess, guesses left: 5
Your Guess: v e n u s

YOU WIN - Good Guess !! the word was : venus

Play again (Y/N):
```

```
        //check if guessed or not after run out of lives
        if( this.guessed )
        {
            System.out.println("Your Guess: " + putInSpaces(this.guessedSoFar.toString()));
            System.out.println("\n YOU WIN - Good Guess !! the word was : " + this.wordToGuess);
        }
        else
        {
            System.out.println("Your Guess: " + putInSpaces(this.guessedSoFar.toString()));
            System.out.println("\n YOU LOOSE, the word was : " + this.wordToGuess);
        }

    }//EOM-play()


    /*=================================================
     * space out the hint words
     *
     *================================================*/
    private String putInSpaces( String aWord )
    {
        StringBuffer sb = new StringBuffer();

        for (int i = 0; i < aWord.length() ; i=i+1)
        {
            sb.append( aWord.charAt(i) + " ");
        }

        return sb.toString();
        //return aWord;
    }

}//EOC
```

Show word with spaces

Java

ASSIGNMENT 2

http://www.projectcartoon.com/cartoon/2

Conor O'Reilly

The code breaker screens

Introduction

The fate of the world rests on your shoulders. A lethal virus is about to be on leashed on the web. You can stop the viruses release if you can guess the sequence of 4 colors that delete the virus.

The possible colors are

R - Red

O - Orange

Y - Yellow

G - Green

B - Blue

I - Indigo

V - Vilot

You have only 8 chances to guess the code. Are you ready to save the world (Y/N)?

Lives: 8

Code: _ _ _ _ Guessed : _ _ _ _ Clues:

Enter a sequence a 4 character sequence from the following values ROYGBIV or 0 to exit:

Lives: 4

Code: _ _ _ _ Guessed : _ _ _ _ Clues:

Code: R _ _ _ Guessed : R O Y G Clues: 1

Code: R _ _ _ Guessed : R B G I Clues: 2

Code: R G _ _ Guessed : R G V I Clues: 2

Code: R G I V Guessed : R G I V Clues: 0

YOU WIN !!

Play again Y/N:

Lives: 0

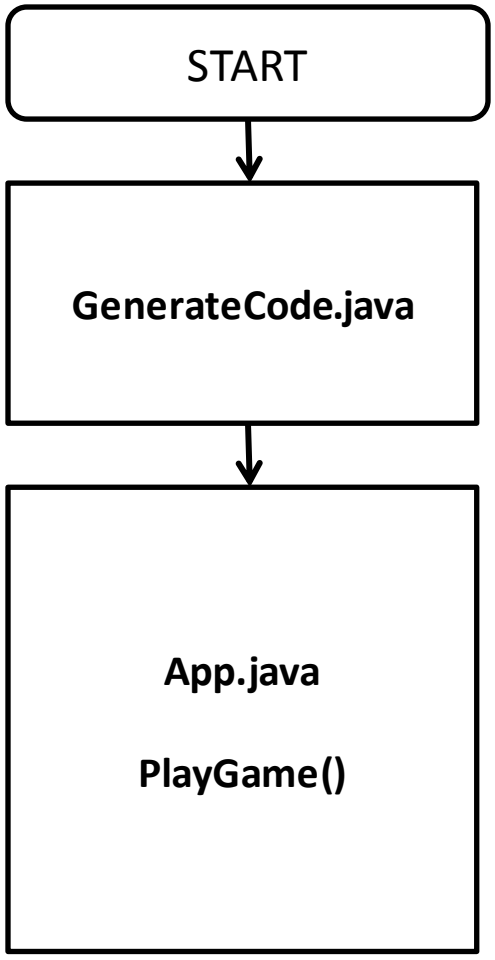
:

:

Code: R _ _ _ Guessed : R V V V Clues: 0

YOU LOOSE, the code was : R G I V

Play again Y/N:



Lives: 8

Code: _ _ _ _ Guessed : _ _ _ _ Clues:

Enter a sequence a 4 character sequence from the following values ROYGBIV or 0 to exit:

Lives: 4

Code: _ _ _ _ Guessed : _ _ _ _ Clues:

Code: R _ _ _ Guessed : R O Y G Clues: 1

Code: R _ _ _ Guessed : R B G I Clues: 2

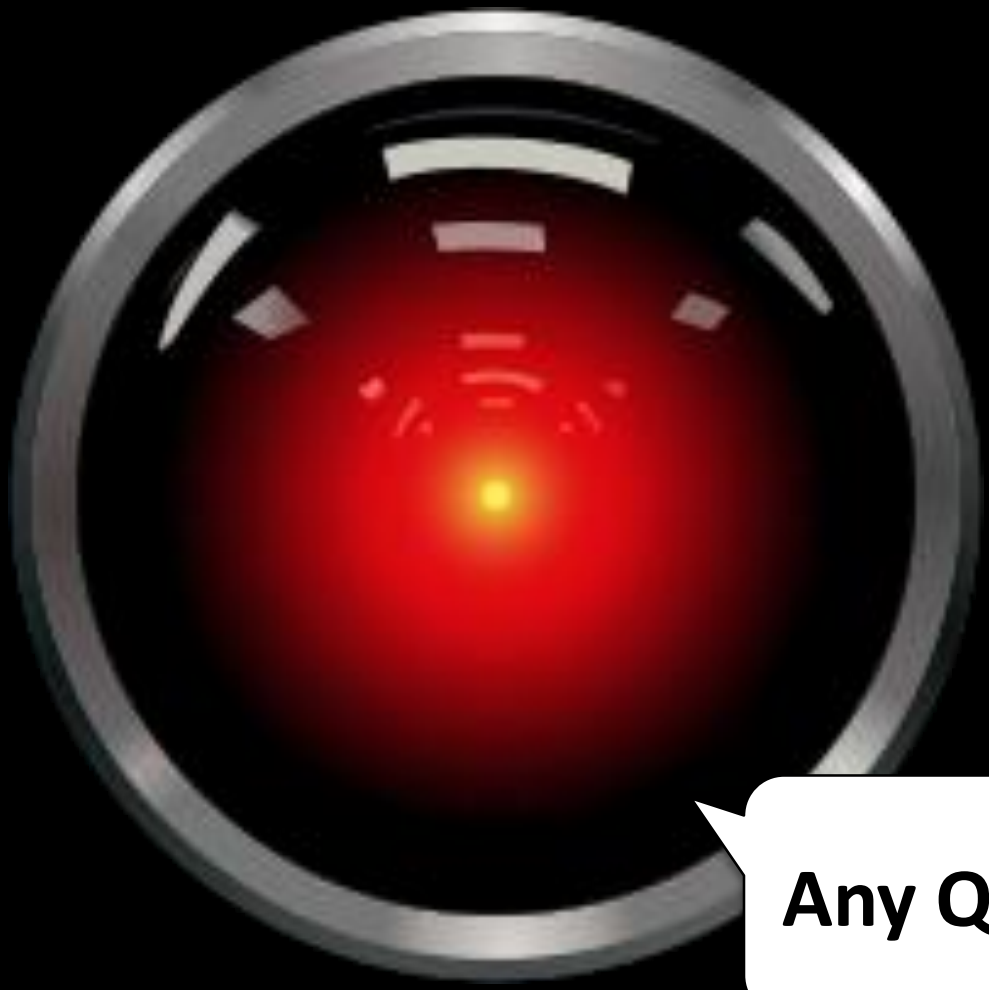
Code: R G _ _ Guessed : R G V I Clues: 2

Code: R G I V Guessed : R G I V Clues: 0

YOU WIN !!

Play again Y/N:

You must use the Javabook classes for data entry and displaying the screens



Any Questions ?

The eLabBook

- Lab Objectives
  - Become familiar with the programming tools and build some basic programs

Problem

- Develop a program that calculates the area of a Rectangle

Problem Definition

Design – Draw object diagram and\or psudo-code

Test case definition

– tests you will use to check the programs functionality

Implementation – translate design into code, do in a set of steps adding additional functionality with each step

---

Test Recording

Code

Screens

Usage documentation

References

To complete the Lab you must show:

1) **Working code**
2) **Completed documentation**
3) **Submit Code & Documentation**

Some problems will have additional levels of difficulty and/or additional exercises

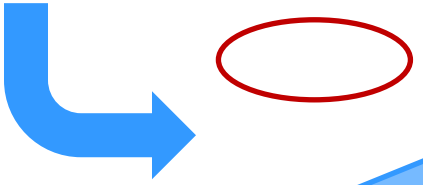
Symbols index

1. steps

- Actions to take, number indicates sequence

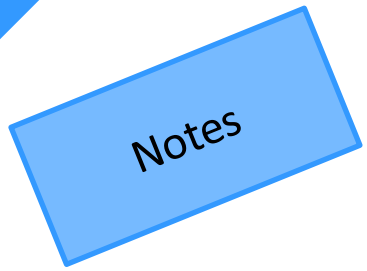
Notes

- Observations and Notes



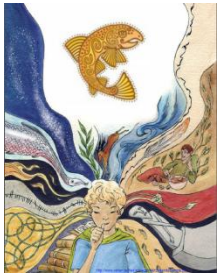
http://

- Clickable URL



[1] [Mayer 2009]

- Indicates the reference



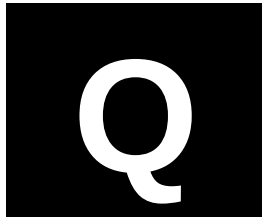
Exam Icon

Local play

Web Play

http://creativecommons.org/licenses/

Ungroup then right click, edit hyperlink on the purple triangle to change the file name of the resource used. The link will become broken of the powerpoint and resource are not in locations relative to each other



Activity Icon

[1] **Author. (2008). Title, Publication**, HTTP://link to resource if possible

Conor O'Reilly