

# Programming Practice

GEEN163

*“There are three roads to ruin; women, gambling and technicians. The most pleasant is with women, the quickest is with gambling, but the surest is with technicians.”*

Georges Pompidou

# Teaching Evaluation

- The official University teaching evaluation survey is on Blackboard
- Complete the survey for ALL classes

# Lab Quiz

- Next week there will be a lab quiz
- You will have to write a program by yourself

# Copy and Paste Error

- Some characters can be represented several different ways in a document, such as quotes and dashes
- Java accepts only the exact character in programs
- The homework assignment PDF has a minus sign that sometimes does not copy correctly
- If Java gives an error “illegal character: \8211”, just retype the dash or quote

# Random Numbers

- Random numbers are useful for simulations and games
- Java provides several pseudo-random number methods
- Pseudo-random numbers appear random, but follow a pattern

# Simple Random Number Class

```
public class RandomProgramming {  
    long seed = 1L;  
    public int nextLong() {  
        seed = (seed * 16807L) % 2147483647L;  
        return (int)seed;  
    }  
}
```

# Random Number Methods

- The static `Math.random()` method returns a random number from 0 to 1.0
- The `Java.util.Random` class provides several methods for random numbers
  - `nextDouble()` – returns a double  $0 \leq x < 1.0$
  - `nextInt()` – returns an int  $> 0$
  - `nextInt(int max)` – returns an int  $0 \leq x < \text{max}$
- These generate uniform random numbers where each number has an equal probability



# Rolling Dice

- Consider a class that simulates dice
- We need a method

```
int rollDice()
```

- that returns an integer from 2 to 12 that is statistically the result of rolling two dice



# How can we simulate rolling two dice?

- A. Create a random number from 1 - 12
- B. Create a random number from 2 to 12
- C. Add two random numbers from 1 - 6
- D. Add two random numbers from 2-12

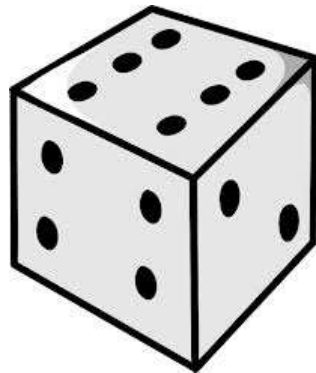
# You Need Two Random Numbers

Two dice totals						
Die 1	Die 2					
1	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

Roll	Probability
2	2.78%
3	5.56%
4	8.33%
5	11.11%
6	13.89%
7	16.67%
8	13.89%
9	11.11%
10	8.33%
11	5.56%
12	2.78%

# Possible Dice Class

```
public class Dice {  
    static java.util.Random rand =  
        new java.util.Random();  
    public static int rollDice() {  
        int die1 = rand.nextInt(6) + 1;  
        int die2 = rand.nextInt(6) + 1;  
        return die1 + die2;  
    }  
}
```



# Static or Non-static

- If a method accesses non-static class instance variables, then the method must be non-static
- If a method does not access any non-static instance variables, it can easily be static
  - Java allows it to be non-static
  - If it does not access object data, there is no reason not to be static

# Craps

- In the dice game Craps, you win your bet if the coming out roll is a 7 or 11. You lose your bet if 2, 3 or 12 are rolled



# Fill in the Blank

```
public class Craps {  
    public static void main(String[] unused) {  
        int money = 100;        // money in your pocket  
        int bet = 10;          // amount of bet  
        int roll = Dice.rollDice();  
  
        // adjust money based on the value of roll  
        // add bet to money if you win  
        // subtract bet from money if you lose  
    }  
}
```

# Possible Solution

```
public class Craps {  
    public static void main(String[] unused) {  
        int money = 100;           // money in your pocket  
        int bet = 10;              // amount of bet  
        int roll = Dice.rollDice();  
  
        if (roll == 7 || roll == 11) {  
            money += bet;  
        } else if (roll == 2 || roll == 3 || roll == 12) {  
            money -= bet;  
        }  
    }  
}
```



# Making Your Point

- In Craps, if the coming out roll is 4, 5, 6, 8, 9 or 10, that becomes your “point”
- You neither win or lose (*yet*)
- You continue to roll until
  - You roll your point, you win your bet and the round is over
  - You roll a 7, you lose your bet and the round is over

# Fill in the blank

```
public class Craps {
    public static void main(String[] unused) {
        int money = 100;
        int bet = 10;
        int roll = Dice.rollDice();
        if (roll == 7 || roll == 11) {
            money += bet;
        }
        if (roll == 2 || roll == 3 || roll == 12) {
            money -= bet;
        }
        int point = roll;
        // Change money based on further rolls until 7
    }
}
```

# Possible Solution

```
boolean endRound = false;
while ( !endRound ) {
    roll = Dice.rollDice();
    if (roll == point) {
        money += bet;
        endRound = true;
    } else if (roll == 7) {
        money -= bet;
        endRound = true;
    }
}
```

# Another Possible Solution

```
while ( true ) {  
    roll = Dice.rollDice();  
    if (roll == point) {  
        money += bet;  
        break;  
    } else if (roll == 7) {  
        money -= bet;  
        break;  
    }  
}
```

# Yet Another Possible Solution

```
int oldMoney = money;
while ( oldMoney == money ) {
    roll = Dice.rollDice();
    if (roll == point) {
        money += bet;
    } else if (roll == 7) {
        money -= bet;
    }
}
```

# Craps Game part 1

```
public class Craps {
    public static void main(String[] args) {
        final int ROUNDS = 250;           // number of rounds
        int point;                         // number to win
        int roll;                           // last roll of the dice
        int money = 100;                    // money left
        int bet = 10;                       // amount of bet
        int counter = 0;                    // number of roundss

        while (counter < ROUNDS && money > 0) {
            counter++;
            roll = Dice.rollDice();        // coming out roll
            System.out.println("Coming out roll of "+roll);
            if (roll == 7 || roll == 11) { // if coming out win
                money += bet;
                System.out.println(" you win $" + bet + " = $" + money);
            } else if (roll == 2 || roll == 3 || roll == 12) {
                money -= bet;
                System.out.println("Craps $" + bet + " giving $" + money);
            } else {
```

# Craps Game part 2

```
point = roll;
comingOut: while ( true ) {
    roll = Dice.rollDice();
    System.out.println("point roll of "+roll);
    if (roll == point) {           // make your point?
        money += bet;             // yes, win the bet
        System.out.println("win $" +bet+" = $" +money);
        break comingOut;
    } else if (roll == 7) {       // crap out?
        money -= bet;             // yes, lose the bet
        System.out.println("7 out $" +bet+" = $" +money);
        break comingOut;
    }
}
}
}
}
System.out.println("done with $" +money);
```

# Is this a good bet?

- How much money, on the average, will we win or lose playing craps?
- Now that we have a program that simulates playing craps, we can use it to play many games and keep track of our winnings



# Computer Simulation

- Our craps game is a computer simulation
- It simulates the playing of more craps games than a human ever has time *or money* to play
- Using computer simulations, an engineer can determine the viability of a design or strategy

If you spend the evening at a casino and play 100 rounds of craps betting \$10 each time, how much money are you likely to have at the end?

A. \$99.86

B. \$86.00

C. \$14.00

D. \$100.14

E. \$114.00

# Calling Methods in Other Classes

- The craps simulation called the rollDice method of the Dice class in two different places
- The rollDice method is static, so it is called on the class instead of an object of the class

```
roll = Dice.rollDice();
```

# Exempt from the Final

- The final exam will be optional for a student when it is determined by the instructor that it is statistically unlikely that the final exam will change the student's grade
- A student always has the option to take the final exam if they wish to do so
- When a student is permitted to not take the final exam, their course grade will be determined by the weighted average of all other graded work

# Statistically Unlikely

- If it is improbable that the final exam will change your grade, there is no reason to take the final
- A student with a 95 average will have to earn an 11 on the final for their grade to drop to a B
- A student with a 75 average needs 103 to get an A and 47 to get a C
- A student with an 81 average might earn less than 71 on the final and get a B. They will have to take the final

# Bank Account Example

- Consider a class that holds information about a bank account
- A bank account object will need to contain
  - name of the account owner
  - balance of the account

# Example Bank Class

```
public class BankAccount {
    private double money;           // balance in the account
    private String owner;          // name of the owner
    // constructor with two parameters
    public BankAccount(double initial, String name){
        money = initial;           // save initial balance
        owner = name;              // save name of owner
    }
    // default constructor
    public BankAccount() {
        money = 0.00;
        owner = "unknown";
    }
}
```

# More Example Methods

```
/* method to add money to the balance */  
public void deposit(double cash) {  
    money = money + cash;  
}
```

```
/* method to remove money from the balance */  
public void withdraw(double cash) {  
    money = money - cash;  
}
```



# Method with an Object Parameter

```
public void transfer(  
    BankAccount you,  
    double amount) {  
  
    you.money -= amount;  
    money     += amount;  
  
}
```

# Write with your team

- Create two objects of the BankAccount class with your name and some initial balance
- Add \$500 to one of the accounts
- Subtract \$1.00 from the other account
- Move \$25 from one account to the other

# Using the BankAccount class

```
BankAccount mine = new
    BankAccount( 10000.00, "Joe" );
BankAccount yours = new
    BankAccount( 47.50, "Fred" );

mine.deposit(500.00);
yours.withdraw( 1.00 );
mine.transfer( yours, 25.00 );
```

# How can you avoid negative balances?

```
public void transfer(  
    BankAccount you,  
    double amount) {  
  
    you.money -= amount;  
    money     += amount;  
  
}
```

# Possible Solution

```
public void transfer(  
    BankAccount you,  
    double amount) {  
  
    if (you.money >= amount) {  
        you.money -= amount;  
        money += amount;  
    }  
}
```

# Schedule

Monday, November 11 <b>Arrays</b> sections 8.3	Wednesday, November 13 <b>More on GUIs</b>	Friday, November 15 <b>Programming practice</b>
Monday, November 18 <b>Programming practice</b>	Wednesday, November 20 <b>Programming practice</b>	Friday, November 22 <b>review</b>
	<b>Lab Quiz</b>	<b>Lab Quiz</b>
Monday, November 25 <b>Exam 3</b>	Wednesday, November 27 <i>Thanksgiving Holiday</i> <i>(no classes)</i>	Friday, November 29 <i>Thanksgiving Holiday</i> <i>(no classes)</i>
Monday, December 2 <b>Software engineering</b>	Wednesday, December 4 <b>review</b>	

# Teaching Evaluation

- The official University teaching evaluation survey is on Blackboard
- Complete the survey for ALL classes

# Lab Quiz

- Next week there will be a lab quiz
- You will have to write a program by yourself