

GK-12 Lesson Plan

Teacher: Steven MacDonald

Period: **Class:** Lawrence High School Statistics Class

Date(s): 2/12/2014

SETTING THE STAGE	
<u>Essential Question</u>	Now that we're getting into more and more complex simulations, how can we develop a strategy to keep the coding aspect simple?
<u>Content Objective(s)</u> (Student-friendly)	Students learn strategies for effective coding, specifically methods for design planning such as skeleton coding and flow chart construction.
<u>Connection to previous or future lessons</u>	Flow charts are introduced so that students have a method for planning out their future project work.
<u>Critical Thinking Questions</u>	
<u>Key Vocabulary</u>	Flow chart, skeleton code
<u>Materials Needed/Safety</u>	Computer, Rstudio
ACTIVE INSTRUCTION	
Launch (Engage)	As an introduction to this lesson, students are shown a simulation of the Powerball lottery as an example of a relatively complex simulation. This introduction is followed by an example of the flowchart used to create the simulation.
Investigation (Explore)	After a lesson on the creation of flow charts, the students are asked to work backwards and create a flowchart based on the previous dice game they created.
TIME FOR REFLECTION	
Summarization (Explain & Extend)	By "reverse-engineering" a flow chart from an existing piece of code, the students gained a more concrete understanding of the creation of flow charts.
Homework	BD questions file attached. Students are asked to create two flow charts for an upcoming program based on a question from their textbook.

GK-12 Lesson Plan

Teacher: Steven MacDonald

```
matches <- 0
nomatches <- 0
n <- 1000
for (i in 1:n){
  ball1 <- sample(0:9,1)
  ball2 <- sample(0:9,1)
  ball3 <- sample(0:9,1)
  if(ball1 == 9 && ball2 == 1 && ball3 == 1){
    match <- match+1
  }else{
    nomatch <- nomatch+1
  }
}
```

```
matches <- 0
nomatches <- 0
n <- 1000
for (i in 1:n){
  day <- sample(1:365,1)
  ball1 <- sample(0:9,1)
  ball2 <- sample(0:9,1)
  ball3 <- sample(0:9,1)

  if(day == 254 && ball1 == 9 && ball2 == 1 && ball3 == 1){
    matches <- matches + 1
  }else
    nomatches <- nomatches+1
}
```

```
result <- vector()
matches <- 0
nomatches <- 0
n <- 1825
zeromonth <- seq(from=1, to=9, by =1)
for (i in 1:n){
  month <- sample(1:12,1)
  if(month == 9 | month == 4 | month == 6 | month == 11){
    day <- sample(1:30,1)
  }else if (month == 2){
    day <- sample(1:28,1)
  }else{
    day <- sample(1:31,1)
  }
}
```

GK-12 Lesson Plan

Teacher: Steven MacDonald

Homework

1) The New York state lottery uses three balls, each drawn from a separate pool of 10 balls numbered from 0 to 9. In order to win this lottery, the drawn numbers must all match **in the same order as the ticket number**.

You need to design a program that simulates n days of this lottery, and records the number of times the winning numbers 9-1-1 are drawn. Sketch a block diagram of your program.

2) On 9/11 2002, the numbers 9-1-1 were drawn in the same lottery. In order to design a program that simulates this event, you now need to take the date into account. Sketch a new block diagram of your new program, keeping in mind that the condition for a match now requires the date to match as well.

GK-12 Lesson Plan

Teacher: Steven MacDonald

