

**Object Oriented  
Java Programming  
Lab Report  
PAUL ERNEST KACHULE  
ID : 2017020501001  
Dec 10, 2019**

- 1) Lottery Simulation**
- 2) Area and circumference of Circle**
- 3) Card game**
- 4) Game of Craps**

# 1) Lottery Simulation

## Review terminology

have seen that in java all program must be defined inside class that contain the main method. This is driver class. We have also some predefined class that can be classified utility class. This contain instance variable.

Class is template that describes an objects data values and method that work with that data. Collectively data and method are referred to as member of the class. There are two categories of data. Those belong to an object and those belong to class. In addition for each data member there is constant and variable data member. Since object is instance of class, variable that belong to object is called instance variable. Each object is created from class as its own version of instance variable. Conversely class variable is belongs to the class. This only one version of class variable shared by all object created from that class.

## Procedure

I wrote a piece of code that implements one class lottery and only main method and declare two array of length six called them Lottery numbers and Random\_Lottery\_Numbers. in addition I also declared the variable count ,random and input store six randomly selected elements between 1 and 49(inclusively) using for loop on the Random\_Lottery\_Numbers .to generate these randomly selected element between 1 and 49 I used the function (int)(Math.Random()\*49+1).and also I used for loop and if statement to avoid the repetition then I promote the user to enter the number between 1 and 49 and store on the second array which named lottery .just like the random generation I avoided the repetition using for loop and if condition to avoid repetition. Restrict the numbers boundary between 1 and 49 means if the user enter out of this bound the compiler doesn't store the value. And ask to enter another number. After that I compare

**both array element one by one. If the same element is found in both array the compiler print out the element and tell the user win and congratulate else the compiler tell user as there is no matched number between user input and randomly selected element. Both selected and inputted element are print as array. THE COMPELETE PROGRAM ATTACHED as source code.**

## Data

**I run the program and observe as follow.**

```
>id
: double
ble
ble
t): void
>id
>id
1
<terminated> Lottery [Java Application] C:\Program Files\IBM\SPSS\Statistics\25\jre\bin\javaw.exe (Dec 14, 2019, 11:22:55 PM)
Welcome to lottery program
Enter your Lottery Number 1 (must be between 1 and 49):
2
Enter your Lottery Number 2 (must be between 1 and 49):
3
Enter your Lottery Number 3 (must be between 1 and 49):
1
Enter your Lottery Number 4 (must be between 1 and 49):
4
Enter your Lottery Number 5 (must be between 1 and 49):
5
Enter your Lottery Number 6 (must be between 1 and 49):
6
YOU WIN!
The numbers that match up are: 3
Numbers you entered : 2 3 1 4 5 6
Selected numbers are : 3 42 22 26 8 10
```

```
imference() : double
ay() : void
adius() : double
() : double
() : double
adius(int) : void
(int) : void
(int) : void
aps.java
iva
<

<terminated> Lottery [Java Application] C:\Program Files\IBM\SPSS\Statistics\25\jre\bin\javaw.exe (Dec 14, 2019, 11:27:00 PM)
23
Enter your Lottery Number 4 (must be between 1 and 49):
12334
invalid number,enter another
2332
invalid number,enter another
2
Enter your Lottery Number 5 (must be between 1 and 49):
134
invalid number,enter another
2
invalid number,enter another

1
Enter your Lottery Number 6 (must be between 1 and 49):
2
invalid number,enter another
4
No matching numbers ... Try Again!
Numbers you entered : 0 49 23 2 1 4
Selected numbers are : 30 32 34 25 9 48
```

## Analysis of data

The result we obtained confirm the lottery simulation law. Definitely confirm the use of for loop, if condition

## Conclusion

Due to effective and save a time one can use the loop and random function to develop different type of games and probability dependent machine work and it is preferred alternative even to save memory.

## 2) Area and circumference of circle

A **class** is a template for **objects**. It defines the **properties** of objects and provides **constructors** for creating objects and methods for manipulating them. A class is also a data type. You can use it to declare **object reference variables**. An object reference variable that appears to hold an object actually contains a reference to that object. Strictly

speaking, an object reference variable and an object are different, but most of the time the distinction can be ignored. An object is an *instance* of a class. You use the “new” operator to create an object, and the *dot operator* (.) to access members of that object through its reference variable. An *instance variable* or *method* belongs to an instance of a class. Its use is associated with individual instances. A *static variable* is a variable shared by all instances of the same class. A *static method* is a method that can be invoked without using instances. Every instance of a class can access the class’s static variables and methods. For clarity, however, it is better to invoke static variables and methods using “ClassName.variable” and “Class Name. Method”. Visibility modifiers specify how the class, method, and data are accessed. A public class, method, or data is accessible to all clients. A “private” method or data is accessible only inside the class. You can provide a getter (accessory) method or a setter (matadors) method to enable clients to see or modify the data. A getter method has the signature “public return Type get Property Name ()”. If the “return Type” is “Boolean”, the get method should be defined as public Boolean is Property Name (). A setter method has the signature “public void set Property Name (data Type property Value)”.

The constructor has exactly the same name as its defining class. Like regular methods, constructors can be overloaded (i.e., multiple constructors can have the same name but different signatures), making it easy to construct objects with different initial data values. Constructors can be overloaded (i.e., multiple constructors can have the same name but different signatures), making it easy to construct objects with different initial data values.

## **Procedure**

I create program implement both technique contain one class with variable x, y and radius. Two constructors one is default constructor and the other is explicitly constructor. The default constructor assign all data element by default the value x, y, radius is (1,1,1).the explicit

constructor has 3 parameter x, y ,r. each parameter have also setter and getter created two method area() calculate and return area ,circumstance which calculate circumstance and return it.to display those method I use a user defined method display().after that I made three circle object c1,c2 using explicit constructor and ,c3 with default constructor in the main method. Then I give c1 small radius c2 larger radius and c3 default radius means 1.then print out all provided data .those are radius, area x, y, area and circumstanes of each circle object.

## **Data**

**circle c1:**

**x= 15, y= 5**

**Radius = 10**

**area= 314.1592653589793**

**circumference=:62.83185307179586**

**circle c2:**

**x= 16, y= 8**

**Radius = 12**

**area= 452.3893421169302**

**circumference=:75.39822368615503**

**circle c3 with default constructor**

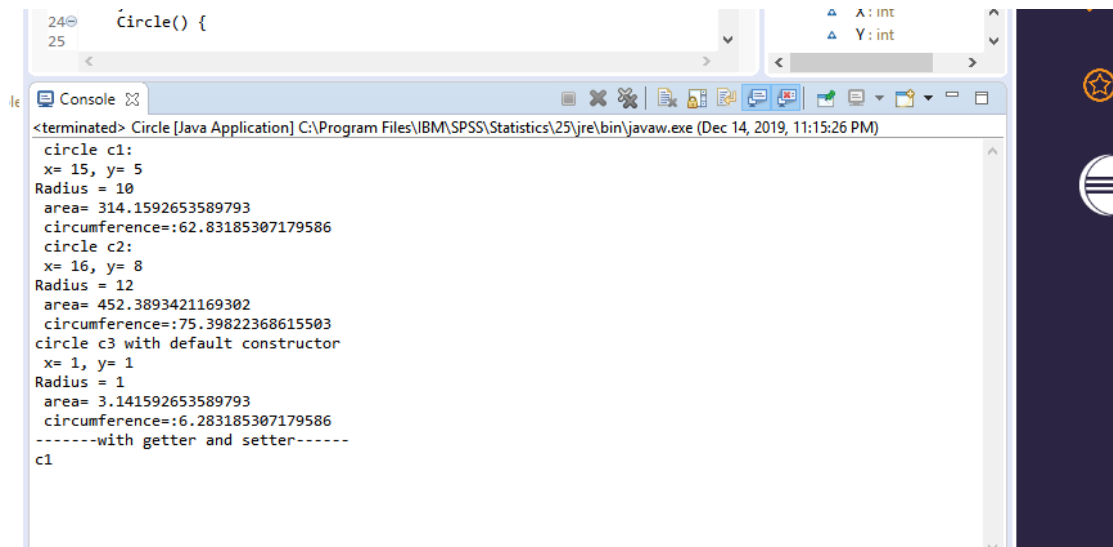
**x= 1, y= 1**

**Radius = 1**

**area= 3.141592653589793**

**circumference=:6.283185307179586**

**-----with getter and setter-----**



```
24 Circle() {
25
<terminated> Circle [Java Application] C:\Program Files\IBM\SPSS\Statistics\25\jre\bin\javaw.exe (Dec 14, 2019, 11:15:26 PM)
circle c1:
x= 15, y= 5
Radius = 10
area= 314.1592653589793
circumference=:62.83185307179586
circle c2:
x= 16, y= 8
Radius = 12
area= 452.3893421169302
circumference=:75.39822368615503
circle c3 with default constructor
x= 1, y= 1
Radius = 1
area= 3.141592653589793
circumference=:6.283185307179586
-----with getter and setter-----
c1
```

**Process finished with exit code 0**

## **Analysis of data**

**When the program runs the result we obtained is what predict in the process. We get result of each circle object based on our design.**

## **Conclusion**

**Due to c organizing and structuring our data based on their similarity and function loop is efficient programming than not oriented programming .surprisingly we can use one data element and method for different object .this avoid the repetition of data and save our time and our memory space. This increase the efficient of our program. Organizing data as such way is useful to store and inherit data and methods.**

### **3) Card game**

First design the class card. This class contain two variable, suit and rank. Both element have the method getter and setter to assign and return value. We have also to string method that return string of suit and rank. Then in the main method we create the game object from card class. In addition create array of length 52 to store all the cards since the card number is 52.also need another two array to store suit and rank separately. Initialize the card on the array name deck. Then shuffle the card using for loop and method (int) (Math. random ()\*deck. Length).

```
for (int i = 0; i < deck.length; i++) {  
    int index = (int)(Math.random() * deck.length);  
    int temp = deck[i];  
    deck[i] = deck[index];  
    deck[index] = temp;  
}
```

Then display the selected card as such

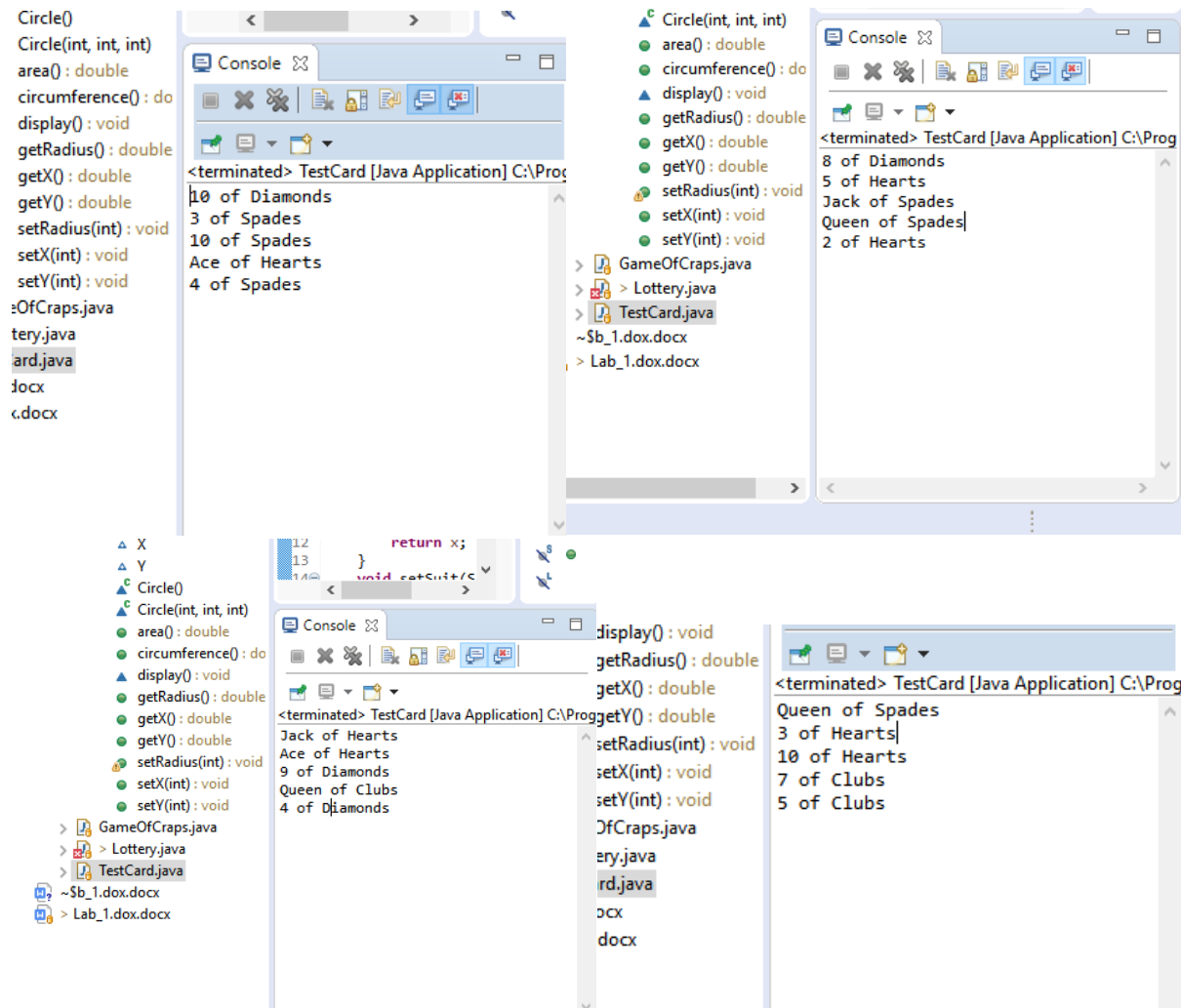
```
for (int i = 0; i < 5; i++) {  
    game.setSuit(suits[deck[i] / 13]);  
    game.setRank(ranks[deck[i] % 13]);  
    System.out.println( game.getRank() + " of " + game.getSuit());  
}
```

Full program will provided

#### **Data**

We run tests 4 times with the following results:





## Result

The result obtained is confirm the designed program. Already get what predict in my design or theory part. This implies that our code is effective.

## Conclusion

**Due to effective way of this coding it is better use to develop different games. Single player can play with this game to enjoy . Organize the loop with OOP is useful and save memory, time and make our program simple to understand.**

## **4) Game of Craps**

**Solution Design: the player is given 1000\$. If he wins the money will increase by the amount of the bet he made. If he lose the money will decrease by the amount of the bet he made. He keeps playing until the money will be 2000 or 0 which is checked by the do while loop. In the playGame() method if the player wins or lose on the first roll it will return 1 or 0 respectively. But if not we will use the while loop to continue rolling the dice until 7 or the first roll is found. If any of these are found the break statement will break out of the loop and 0 or 1 is returned in the same way as before. The makeBet() method will check whether the player's bet is less than his money and greater than 1. If then the player can make the bet. If not the player will be asked to bet again. The method tossDice() in the class Dice returns the sum of the results from rolling two dice. It is declared as static in order to get access to it through class name.**

**Lessons Learned:**

- **Accessing static method through class**
- **Flow control**
- **Putting limitation on user's input**
- **Loops**