

Object Orientated Applications Coursework

Sean Savitz – C1673010

Question 1)

- i) My code is found in the appendix and attached in the ZIP file.
- ii) My UML class diagram representing the program which results from my modifications

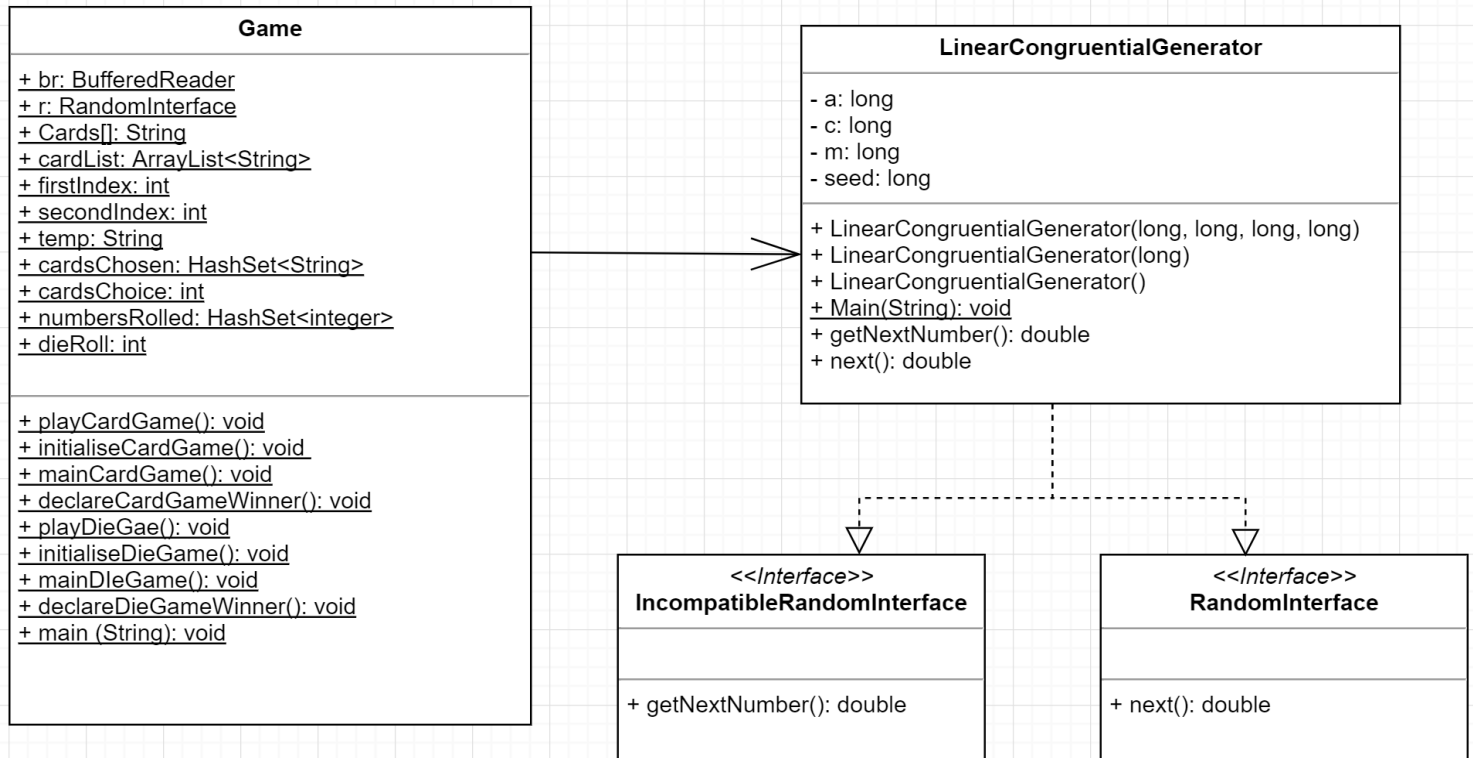


Figure 1 - Underlining of methods and attributes represents that they are static.

- iii) When the 'Game.java' file is run, the user chooses whether they want to play a Dice or Card game. The main method requires a string input where the user can either input a 'c' to initialise the Card game or a 'd' to initialise the Dice game. If neither 'c' or 'd' is inputted then the programme will break. In the card game, a random number generator is used to select two cards randomly out of an array which holds all of the cards in the deck. If an ace is selected out of the deck of cards then the user will win, otherwise the user will lose. In either case, the game ends. Three methods are called when the card game is chosen; that of 'initialiseCardGame()' which begins the game, 'mainCardGame()' which displays the two cards that were chosen and 'declareCardGameWinner()' which presents whether the user has won or lost the game.

In the dice game, a random number generator is used to roll two numbers between 1 and 6. If either one of the number is a 1 then the user will win. If neither of the two numbers is a 1 then the user will lose. Three functions are called when the die game is chosen; that of 'initialiseDieGame()', 'mainDieGame()' and 'declareDieGameWinner()' which have similar functionalities as the card game.

Both of these games are dealt with in the 'Game' class which is inelegant and lacks cohesion. The random number generator for each game is found in the 'LinearCongruentialGenerator' class.

Question 2)

i) In Question 1, I found that the dice and card games were both present in the 'Game' class. In order to ensure that my program has a high cohesion, I will need to separate the two games as well as use other classes such as a Factory. I need to make sure that the classes that I create have minimal dependency on one another to ensure that my program has low coupling. The following classes are suitable to be incorporated in my improved version of the program.

- **Abstract** – The abstract class will declare all of the important game methods which must be present in the CardGame and DiceGame classes. The CardGame and DiceGame classes will extend to the abstract class. The following are the methods that will be implemented.
 - startGame() – This will initiate the game that will be played.
 - gameIntro() – This will provide an introduction to each of the games.
 - getText() – This will provide the text declaring whether the user has won.
 - quickWin() – Used so that if the user wins in the dice game or gets an ace in the card game on their first roll, the game will terminate and the user will win.
 - randomGenerator() – This passes on the random number that has been generated to either the card or dice game.
- **Factory** – The Factory class will contain an IF statement to ensure that only a 'c' to indicate the card game or a 'd' to indicate the dice game is inputted by the user when the command line asks what game the user wants to play. If there is any other type of input, an error statement is printed, and the game will terminate.
- **DiceGame** – The DiceGame class will contain an introduction to the game. The user will 'roll' the dice and a random number will be generated between 1 and 6. There will be an appropriate output, changing whether the user wins by rolling a 1 or loses.
- **CardGame** – The CardGame class will contain an introduction to the game. The user will be given two cards and the rest of the cards of the deck will be displayed. There will be an appropriate output, changing whether the user wins by picking an Ace or loses.
- **Game** – The game class will contain the main method which asks the user to input either a 'c' or a 'd'. This will be received from the Factory class, initiating either the CardGame class or the DiceGame class depending on the user's choice.
- **UserInput** – The UserInput class will use a BufferedReader to read the input of the user to be used throughout the program.

The 'LinearCongruentialGenerator' class stays the same as question 1 and is used to create a random number which is used in the Dice Game and Card Game. The factory class is an important implementation as this allows for additional flexibility for change, especially in the future when more games may be included into the program.

2ii) UML class diagram representing the improved program.

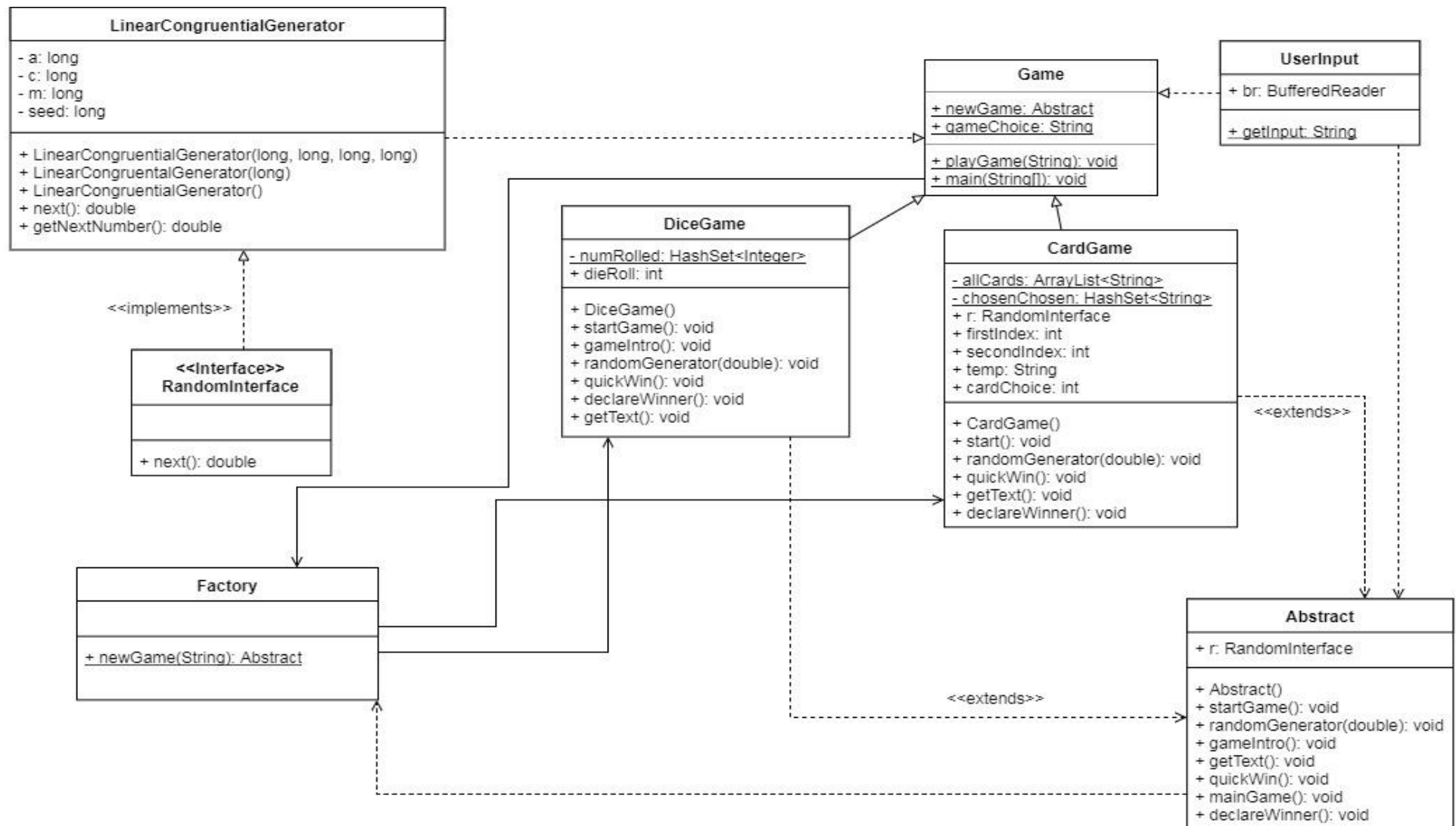


Figure 2 - Underlining of methods and attributes represents that they are static.

Appendix

Question 1:

All Changes have been marked with a yellow text highlighter:

LinearCongruentialGenerator.java

```
public class LinearCongruentialGenerator implements IncompatibleRandomInterface,
RandomInterface {
    // Generates pseudo-random numbers using:
    //  $X(n+1) = (aX(n) + c) \pmod{m}$ 
    // for suitable a, c and m. The numbers are "normalised" to the range
    // [0, 1) by computing  $X(n+1) / m$ .

    private long a, c, m, seed;
    // Need to be long in order to hold typical values...

    public LinearCongruentialGenerator(long a_value, long c_value, long m_value, long
s_value) {
        a=a_value; c=c_value; m=m_value; seed=s_value;
    }

    public LinearCongruentialGenerator(long seed) {
        // Set a, c and m to values suggested in Press, Teukolsky, et al., "Numerical
Recipes"
        this(1664525, 1013904223, 4294967296L, seed);
        // NB "L" on the end is the way that a long integer can be specified. The
        // smaller ones are type-cast silently to longs, but the large number is too
        // big to fit into an ordinary int, so needs to be defined explicitly
    }

    public LinearCongruentialGenerator() {
        // (Re-)set seed to an arbitrary value, having first constructed the object using
        // zero as the seed. The point is that we don't know what m is until after it has
        // been initialised.

        this(0); seed=System.currentTimeMillis() % m;
    }

    public static void main(String args[]) {
        // Just a little bit of test code, to illustrate use of this class.
        IncompatibleRandomInterface r=new LinearCongruentialGenerator();
        for (int i=0; i<10; i++) System.out.println(r.getNextNumber());

        // Since RandomInterface doesn't know about the instance variables defined in
this
        // particular implementation, LinearCongruentialGenerator, we need to type-cast
        // in order to print out the parameters (primarily for "debugging" purposes).

        LinearCongruentialGenerator temp=(LinearCongruentialGenerator) r;
        System.out.println("a: " + temp.a + " c: " + temp.c + " m: " + temp.m + "
seed: " + temp.seed);
    }

    public double getNextNumber() {
        seed = (a * seed + c) % m;
        return (double) seed/m;
    }

    public double next(){
```

```
    return getNextNumber();  
}  
  
}
```

Question 2)

Evidence of Card Game being played:

```
CA: C:\Windows\system32\cmd.exe - java Game

C:\Users\c1673010\Desktop\question2>java Game
Type (c) to play the Card game and (d) to play the Die game! c
15Spds, 7Clbs, 10Dmnds, 2Spds, 4Dmnds, KDmnds, QHrts, JClbs, KClbs, AHrts, 9Dmnd
s, 8Clbs, 6Spds, 10Clbs, 2Clbs, 9Spds, ASpds, AC1bs, 10Hrts, 7Hrts, JHrts, 4Spds
, KHrts, 6Dmnds, 8Dmnds, 2Hrts, JDmnds, 8Hrts, 6Clbs, 3Clbs, 2Dmnds, 3Dmnds, 3Sp
ds, JSpds, 5Hrts, 7Spds, 5Dmnds, 8Spds, 4Hrts, 6Hrts, 3Hrts, QClbs, 9Hrts, KSpds
, 4Clbs, ADmnds, 5Clbs, 10Spds, 7Dmnds, 9Clbs, QDmnds, QSpds1

-----Welcome to the Card Game!-----
Get an Ace to win!
Press <RETURN> to choose a card
_
```

Screenshot showing when the Card game is initialised

```
CA: C:\Windows\system32\cmd.exe

Type (c) to play the Card game and (d) to play the Die game! c
15Spds, 7Clbs, 10Dmnds, 2Spds, 4Dmnds, KDmnds, QHrts, JClbs, KClbs, AHrts, 9Dmnd
s, 8Clbs, 6Spds, 10Clbs, 2Clbs, 9Spds, ASpds, AC1bs, 10Hrts, 7Hrts, JHrts, 4Spds
, KHrts, 6Dmnds, 8Dmnds, 2Hrts, JDmnds, 8Hrts, 6Clbs, 3Clbs, 2Dmnds, 3Dmnds, 3Sp
ds, JSpds, 5Hrts, 7Spds, 5Dmnds, 8Spds, 4Hrts, 6Hrts, 3Hrts, QClbs, 9Hrts, KSpds
, 4Clbs, ADmnds, 5Clbs, 10Spds, 7Dmnds, 9Clbs, QDmnds, QSpds1

-----Welcome to the Card Game!-----
Get an Ace to win!
Press <RETURN> to choose a card

You chose 7Dmnds
Press <RETURN> to choose a card

You chose 5Clbs
The Cards that have been chosen are: [5Clbs, 7Dmnds]
The Remaining cards are: [15Spds, 7Clbs, 10Dmnds, 2Spds, 4Dmnds, KDmnds, QHrts, J
Clbs, KClbs, AHrts, 9Dmnds, 8Clbs, 6Spds, 10Clbs, 2Clbs, 9Spds, ASpds, AC1bs, 10
Hrts, 7Hrts, JHrts, 4Spds, KHrts, 6Dmnds, 8Dmnds, 2Hrts, JDmnds, 8Hrts, 6Clbs, 3
Clbs, 2Dmnds, 3Dmnds, 3Spds, JSpds, 5Hrts, 7Spds, 5Dmnds, 8Spds, 4Hrts, 6Hrts, 3
Hrts, QClbs, 9Hrts, KSpds, 4Clbs, ADmnds, 10Spds, 9Clbs, QDmnds, QSpds1

Sorry, you have lost!

C:\Users\c1673010\Desktop\question2>
```

When the user does not pick an Ace and loses the game.

```
CA: C:\Windows\system32\cmd.exe

Sorry, you have lost!

C:\Users\c1673010\Desktop\question2>java Game
Type (c) to play the Card game and (d) to play the Die game! c
18Spds, 5Dmnds, 10Spds, KDmnds, 5Hrts, 6Dmnds, QClbs, 4Dmnds, 3Spds, 9Dmnds, JHr
ts, AC1bs, 9Spds, 2Dmnds, 10Hrts, QDmnds, 7Dmnds, 6Hrts, 3Hrts, KClbs, JDmnds, 9
Hrts, 10Dmnds, 3Clbs, 8Hrts, 7Hrts, 2Spds, 9Clbs, 3Dmnds, AHrts, 7Spds, KSpds, 8
Clbs, ASpds, 2Hrts, 10Clbs, 4Hrts, JSpds, KHrts, QSpds, 7Clbs, QHrts, JClbs, 4Sp
ds, 6Clbs, 5Clbs, 6Spds, 8Dmnds, ADmnds, 4Clbs, 5Spds, 2Clbs1

-----Welcome to the Card Game!-----
Get an Ace to win!
Press <RETURN> to choose a card

You chose AC1bs
The Cards that have been chosen are: [AC1bs]
The Remaining cards are: [18Spds, 5Dmnds, 10Spds, KDmnds, 5Hrts, 6Dmnds, QClbs, 4
Dmnds, 3Spds, 9Dmnds, JHrts, 9Spds, 2Dmnds, 10Hrts, QDmnds, 7Dmnds, 6Hrts, 3Hrts
, KClbs, JDmnds, 9Hrts, 10Dmnds, 3Clbs, 8Hrts, 7Hrts, 2Spds, 9Clbs, 3Dmnds, AHrt
s, 7Spds, KSpds, 8Clbs, ASpds, 2Hrts, 10Clbs, 4Hrts, JSpds, KHrts, QSpds, 7Clbs,
QHrts, JClbs, 4Spds, 6Clbs, 5Clbs, 6Spds, 8Dmnds, ADmnds, 4Clbs, 5Spds, 2Clbs1

CONGRATULATIONS YOU HAVE WON!

C:\Users\c1673010\Desktop\question2>
```

When the user gets an Ace and, therefore, wins the game.

```

C:\Windows\system32\cmd.exe
Sorry, you have lost!

C:\Users\c1673010\Desktop\question2>java Game
Type (c) to play the Card game and (d) to play the Die game! c
[QClbs, ADmnds, 6Clbs, 9Hrts, 10Clbs, 3Dmnds, 10Spds, 2Hrts, KSpds, 7Spds, AClbs,
QHrts, 2Dmnds, JSpds, 8Dmnds, 6Hrts, AHrts, 9Spds, KDmnds, KClbs, 6Spds, 4Spds,
5Spds, JDmnds, 10Dmnds, 3Hrts, 4Dmnds, 8Clbs, 5Clbs, 7Hrts, QDmnds, 4Hrts, ASpds,
2Spds, 7Dmnds, 4Clbs, KHrts, 7Clbs, JHrts, JClbs, 3Clbs, 10Hrts, 3Spds, 8Spds,
9Clbs, 2Clbs, QSpds, 9Dmnds, 5Hrts, 6Dmnds, 5Dmnds, 8Hrts]

----Welcome to the Card Game!----
Get an Ace to win!
Press <RETURN> to choose a card

You chose ADmnds
The Cards that have been chosen are: [ADmnds]
The Remaining cards are: [QClbs, 6Clbs, 9Hrts, 10Clbs, 3Dmnds, 10Spds, 2Hrts, KS
pds, 7Spds, AClbs, QHrts, 2Dmnds, JSpds, 8Dmnds, 6Hrts, AHrts, 9Spds, KDmnds, KC
lbs, 6Spds, 4Spds, 5Spds, JDmnds, 10Dmnds, 3Hrts, 4Dmnds, 8Clbs, 5Clbs, 7Hrts, Q
Dmnds, 4Hrts, ASpds, 2Spds, 7Dmnds, 4Clbs, KHrts, 7Clbs, JHrts, JClbs, 3Clbs, 10
Hrts, 3Spds, 8Spds, 9Clbs, 2Clbs, QSpds, 9Dmnds, 5Hrts, 6Dmnds, 5Dmnds, 8Hrts]

CONGRATULATIONS YOU HAVE WON!

C:\Users\c1673010\Desktop\question2>_

```

Added functionality where if the user gets an ace on first turn, game will terminate and the user will win.

Evidence of Dice game being played:

```

C:\Windows\system32\cmd.exe - java Game
. 5Spds, JDmnds, 10Dmnds, 3Hrts, 4Dmnds, 8Clbs, 5Clbs, 7Hrts, QDmnds, 4Hrts, ASpds,
2Spds, 7Dmnds, 4Clbs, KHrts, 7Clbs, JHrts, JClbs, 3Clbs, 10Hrts, 3Spds, 8Spds,
9Clbs, 2Clbs, QSpds, 9Dmnds, 5Hrts, 6Dmnds, 5Dmnds, 8Hrts]

----Welcome to the Card Game!----
Get an Ace to win!
Press <RETURN> to choose a card

You chose ADmnds
The Cards that have been chosen are: [ADmnds]
The Remaining cards are: [QClbs, 6Clbs, 9Hrts, 10Clbs, 3Dmnds, 10Spds, 2Hrts, KS
pds, 7Spds, AClbs, QHrts, 2Dmnds, JSpds, 8Dmnds, 6Hrts, AHrts, 9Spds, KDmnds, KC
lbs, 6Spds, 4Spds, 5Spds, JDmnds, 10Dmnds, 3Hrts, 4Dmnds, 8Clbs, 5Clbs, 7Hrts, Q
Dmnds, 4Hrts, ASpds, 2Spds, 7Dmnds, 4Clbs, KHrts, 7Clbs, JHrts, JClbs, 3Clbs, 10
Hrts, 3Spds, 8Spds, 9Clbs, 2Clbs, QSpds, 9Dmnds, 5Hrts, 6Dmnds, 5Dmnds, 8Hrts]

CONGRATULATIONS YOU HAVE WON!

C:\Users\c1673010\Desktop\question2>java Game
Type (c) to play the Card game and (d) to play the Die game! d

----Welcome to the Dice Game!----
Roll a 1 to win!
Press <RETURN> to roll the die

```

Screenshot showing when the Dice game is initialised

```

C:\Windows\system32\cmd.exe
You chose ADmnds
The Cards that have been chosen are: [ADmnds]
The Remaining cards are: [QClbs, 6Clbs, 9Hrts, 10Clbs, 3Dmnds, 10Spds, 2Hrts, KS
pds, 7Spds, AClbs, QHrts, 2Dmnds, JSpds, 8Dmnds, 6Hrts, AHrts, 9Spds, KDmnds, KC
lbs, 6Spds, 4Spds, 5Spds, JDmnds, 10Dmnds, 3Hrts, 4Dmnds, 8Clbs, 5Clbs, 7Hrts, Q
Dmnds, 4Hrts, ASpds, 2Spds, 7Dmnds, 4Clbs, KHrts, 7Clbs, JHrts, JClbs, 3Clbs, 10
Hrts, 3Spds, 8Spds, 9Clbs, 2Clbs, QSpds, 9Dmnds, 5Hrts, 6Dmnds, 5Dmnds, 8Hrts]

CONGRATULATIONS YOU HAVE WON!

C:\Users\c1673010\Desktop\question2>java Game
Type (c) to play the Card game and (d) to play the Die game! d

----Welcome to the Dice Game!----
Roll a 1 to win!
Press <RETURN> to roll the die

You rolled 6
Press <RETURN> to roll the die

You rolled 6

Sorry, you have lost!

C:\Users\c1673010\Desktop\question2>_

```

When the user does not roll a 1 and loses the game.

```
C:\Windows\system32\cmd.exe

You rolled 2
Press <RETURN> to roll the die

You rolled 4

Sorry, you have lost!

C:\Users\c1673010\Desktop\question2>
C:\Users\c1673010\Desktop\question2>
C:\Users\c1673010\Desktop\question2>java Game
Type <c> to play the Card game and <d> to play the Die game! d

-----Welcome to the Dice Game!-----
Roll a 1 to win!
Press <RETURN> to roll the die

You rolled 3
Press <RETURN> to roll the die

You rolled 1

CONGRATULATIONS YOU HAVE WON!

C:\Users\c1673010\Desktop\question2>
```

When the user rolls a 1 and, therefore, wins the game.

```
C:\Windows\system32\cmd.exe

Type <c> to play the Card game and <d> to play the Die game! d

-----Welcome to the Dice Game!-----
Roll a 1 to win!
Press <RETURN> to roll the die

You rolled 5
Press <RETURN> to roll the die

You rolled 6

Sorry, you have lost!

C:\Users\c1673010\Desktop\question2>java Game
Type <c> to play the Card game and <d> to play the Die game! d

-----Welcome to the Dice Game!-----
Roll a 1 to win!
Press <RETURN> to roll the die

You rolled 1

CONGRATULATIONS YOU HAVE WON!

C:\Users\c1673010\Desktop\question2>
```

Added functionality where if the user rolls a 1 on first turn, game will terminate and the user will win.

Added functionality where the game terminates when another character is written instead of 'c' or 'd'.

```
C:\Windows\system32\cmd.exe

Sorry, you have lost!

C:\Users\c1673010\Desktop\question2>java Game
Type <c> to play the Card game and <d> to play the Die game! d

-----Welcome to the Dice Game!-----
Roll a 1 to win!
Press <RETURN> to roll the die

You rolled 4
Press <RETURN> to roll the die

You rolled 1

CONGRATULATIONS YOU HAVE WON!

C:\Users\c1673010\Desktop\question2>
C:\Users\c1673010\Desktop\question2>java Game
Type <c> to play the Card game and <d> to play the Die game! f

    There is an Error!

Exiting Game

C:\Users\c1673010\Desktop\question2>
```


Source Code for Question 2

Abstract.java

```
abstract class Abstract {

    public Abstract() {

    }

    abstract void startGame() throws Exception;

    // Allows the random number to be passed to the either card or die game
    abstract void randomGenerator(double randomChoice);

    abstract void gameIntro(); //intro for the different games

    abstract void getText();//provides text for winning/losing

    abstract void quickWin();

    // allows for intro of game, random number generated and two rolls of dice or
two cards to be picked
    public void mainGame() throws Exception {
        RandomInterface r=new LinearCongruentialGenerator();

        gameIntro();

        for (int i=0; i<2; i++) {
            getText();
            UserInput.getInput();
            double randomChoice = r.next();
            randomGenerator(randomChoice);
            quickWin();
        }
    }

    abstract void declareWinner();
}
```

CardGame.java

```
import java.io.*;
import java.util.*;

class CardGame extends Abstract {
    private static ArrayList<String> allCards;
    private static HashSet<String> chosenCards;
    RandomInterface r=new LinearCongruentialGenerator();

    public CardGame() {
        this.chosenCards = new HashSet<String>();
    }

    void gameIntro(){
        System.out.println("");
        System.out.println("-----Welcome to the Card Game!-----");
        System.out.println("Get an Ace to win!");
    }
}
```

```

void startGame() throws Exception {
    // The initialisation phase:
    // Create a list of cards ... and shuffle them
    String cards[]={ "AHrts", "2Hrts", "3Hrts", "4Hrts", "5Hrts", "6Hrts",
        "7Hrts", "8Hrts", "9Hrts", "10Hrts", "JHrts",
        "QHrts", "KHrts",
        "ADmnds", "2Dmnds", "3Dmnds", "4Dmnds", "5Dmnds",
        "6Dmnds", "7Dmnds", "8Dmnds", "9Dmnds", "10Dmnds",
        "JDmnds", "QDmnds", "KDmnds",
        "ASpds", "2Spds", "3Spds", "4Spds", "5Spds", "6Spds",
        "7Spds", "8Spds", "9Spds", "10Spds", "JSpds",
        "QSpds", "KSpds",
        "AClbs", "2Clbs", "3Clbs", "4Clbs", "5Clbs", "6Clbs",
        "7Clbs", "8Clbs", "9Clbs", "10Clbs", "JClbs",
        "QClbs", "KClbs"};
    this.allCards = new ArrayList<String> (Arrays.asList(cards));
    // Taking advantage of "generics" to tell the compiler all the elements
will be Strings

    // cards are shuffled
    for (int i=0; i<100; i++) {
        // choose two random cards at random
        int firstIndex=((int) (r.next() * 52));
        int secondIndex=((int) (r.next() * 52));
        String temp=(String) allCards.get(firstIndex);
        allCards.set(firstIndex, allCards.get(secondIndex));
        allCards.set(secondIndex, temp);
    }

    // Print rest of cards
    System.out.println(allCards);
}

void randomGenerator(double randomChoice) {
    int cardChoice= (int)(randomChoice * allCards.size());
    System.out.println("You chose " + allCards.get(cardChoice));
    chosenCards.add(allCards.remove(cardChoice));
}

void getText() {
    System.out.println("Press <RETURN> to choose a card");
}

void quickWin(){
    if (chosenCards.contains("AHrts") || chosenCards.contains("ADmnds") ||
        chosenCards.contains("ASpds") || chosenCards.contains("AClbs")){
        declareWinner();
        System.exit(0);
    }
}

void declareWinner() {

    // User wins if one of them is an Ace
    System.out.println("The Cards that have been chosen are: " + chosenCards);
    System.out.println("The Remaining cards are: " + allCards);

    if (chosenCards.contains("AHrts") || chosenCards.contains("ADmnds") ||
        chosenCards.contains("ASpds") || chosenCards.contains("AClbs")) {
        System.out.println("\nCONGRATULATIONS YOU HAVE WON!");
    }
    else System.out.println("\nSorry, you have lost!");
}

```

```
}  
}
```

DiceGame.java

```
import java.io.*;  
import java.util.*;  
  
class DiceGame extends Abstract {  
    private static HashSet<Integer> numRolled;  
  
    public DiceGame() {  
        this.numRolled = new HashSet<Integer>();  
    }  
  
    void startGame() throws Exception {  
        return;  
    }  
  
    void gameIntro() {  
        System.out.println("");  
        System.out.println("-----Welcome to the Dice Game!-----");  
        System.out.println("Roll a 1 to win!");  
    }  
  
    void randomGenerator(double randomChoice) {  
        int dieRoll = (int)(randomChoice * 6) + 1;  
  
        System.out.println("You rolled " + dieRoll);  
        numRolled.add(new Integer(dieRoll));  
    }  
  
    void quickWin(){ //used so that if 1 is generated on first turn, the game ends  
and user wins  
        if (numRolled.contains(new Integer(1))){  
            declareWinner();  
            System.exit(0);  
        }  
    }  
  
    void declareWinner() { //used to find out whether the user has won or not  
        // if number rolled is 1 then user wins, if not then user loses  
        if (numRolled.contains(new Integer(1))) {  
            System.out.println("\nCONGRATULATIONS YOU HAVE WON!");  
        }  
        else System.out.println("\nSorry, you have lost!");  
    }  
  
    void getText() {  
        System.out.println("Press <RETURN> to roll the die");  
    }  
}
```

Factory.java

```
import java.io.*;  
import java.util.*;  
  
class DiceGame extends Abstract {  
    private static HashSet<Integer> numRolled;
```

```

public DiceGame() {
    this.numRolled = new HashSet<Integer>();
}

void startGame() throws Exception {
    return;
}

void gameIntro() {
    System.out.println("");
    System.out.println("-----Welcome to the Dice Game!-----");
    System.out.println("Roll a 1 to win!");
}

void randomGenerator(double randomChoice) {
    int dieRoll = (int)(randomChoice * 6) + 1;

    System.out.println("You rolled " + dieRoll);
    numRolled.add(new Integer(dieRoll));
}

void quickWin(){ //used so that if 1 is generated on first turn, the game ends
and user wins
    if (numRolled.contains(new Integer(1))){
        declareWinner();
        System.exit(0);
    }
}

void declareWinner() { //used to find out whether the user has won or not
// if number rolled is 1 then user wins, if not then user loses
if (numRolled.contains(new Integer(1))) {
    System.out.println("\nCONGRATULATIONS YOU HAVE WON!");
}
else System.out.println("\nSorry, you have lost!");
}

void getText() {
    System.out.println("Press <RETURN> to roll the die");
}
}

```

Game.java

```

import java.io.*;
import java.util.*;

public class Game {

    //begins game
    public static void playGame(String game) throws Exception {
        //Gets input from 'Factory.java' - either card or die game
        Abstract newGame = Factory.newGame(game);
        newGame.startGame();
        newGame.mainGame();
        newGame.declareWinner();
    }

    public static void main (String[] args) throws Exception { //main function, in-
initiated when the file is run
        System.out.print("Type (c) to play the Card game and (d) to play the Die
game! ");

```

```

        String gameChoice = UserInput.getInput();
        playGame(gameChoice);
    }
}

```

LinearCongruentialGenerator.java

```

public class LinearCongruentialGenerator implements RandomInterface {
    // Generates pseudo-random numbers using:
    //  $X(n+1) = (aX(n) + c) \pmod{m}$ 
    // for suitable a, c and m. The numbers are "normalised" to the range
    // [0, 1) by computing  $X(n+1) / m$ .

    private long a, c, m, seed;
    // Need to be long in order to hold typical values ...

    public LinearCongruentialGenerator(long a_value, long c_value, long m_value, long
s_value) {
        a=a_value; c=c_value; m=m_value; seed=s_value;
    }

    public LinearCongruentialGenerator(long seed) {
        // Set a, c and m to values suggested in Press, Teukolsky, et al., "Numerical
Recipies"
        this(1664525, 1013904223, 4294967296L, seed);
        // NB "L" on the end is the way that a long integer can be specified. The
        // smaller ones are type-cast silently to longs, but the large number is too
        // big to fit into an ordinary int, so needs to be defined explicitly
    }

    public LinearCongruentialGenerator() {
        // (Re-)set seed to an arbitrary value, having first constructed the object using
        // zero as the seed. The point is that we don't know what m is until after it has
        // been initialised.

        this(0); seed=System.currentTimeMillis() % m;

    }

    public double getNextNumber() {
        seed = (a * seed + c) % m;
        return (double) seed/m;
    }

    public double next(){
        return getNextNumber();
    }
}

```

RandomInterface.java

```

public interface RandomInterface {
    // Simply defines a method for retrieving the next random number
    double next();
}

```

UserInput.java

```
import java.io.*;
import java.util.*;

public class UserInput {
    // Used to get input from user to be used throughout program
    public static String getInput() throws Exception {
        // The BufferedReader used throughout
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        return br.readLine();
    }
}
```