# Lab 9\_2 - Inheritance

# Part 1

1. Start a Java project named **Lab9\_2A**
2. Create a secondary class named **DiceGame\_6**. It should have the following:
   1. 3 protected int instance variables to hold the rolls for a first, second & third dice
   2. A protected int instance variable to hold the game score
   3. A no argument constructor that sets the instance variables to 0
   4. An int method that will return a random number between 1 and 6 (You’ll call this to roll one die.)
   5. A void method that draws all 3 dice at once (side by side). It should look like this example (with the appropriate values from each die.)

Picture of line drawings for 3 dice

* 1. A void method that plays the game. It will use the method from step #d for each dice to roll it (by setting each die = the returned value from that method) Then it will use the method from step #e to draw all 3 dice at once.

If all 3 dice are the same, it will print that the player won and stop.

If only 2 dice are the same, it will add 5 to the game score.

If the score is >= 20, the player wins. This should be printed, and the game stopped.

If the first dice < the second die and the second die < the third die, the player automatically loses. This should be printed, and the game stopped.

It will continue rolling until a win or loss happens.

Print the score and a blank line after each set of rolls.

* 1. Make all methods public. They will not have parameters unless the instructions specifically say so.

1. Create a new class named **DiceGame\_10** and make it a child (subclass) of **DiceGame\_6**. It will use 10-sided dice and should have the following:
   1. A method overriding the #2d method above to roll a die. It should return a random number between 1 and 10 instead.
   2. A method overriding the #2f method above. It will do much the same thing, but change the rules for winning and losing to:

If all 3 dice values are divisible by 5 (i.e. – a 5 or a 10) then it is an automatic win.

If all 3 dice have values > 5, then the player automatically loses.

If only 2 dice are divisible by 5, then add 5 to the game score.

If the score is >= 15, the player wins.

1. Back in the main method of the main class, create a **DiceGame\_6** object and then run the method to play the game for it.
2. Do the same thing with a **DiceGame\_10** object.
3. Print a statement before you run each game to say which type of object it is and print a blank line between the two games.

# Part 2

1. Create a new Java project called **Lab9\_2B**
2. Create an abstract class named **Customer**. It should have:
   1. Protected instance variables to hold the customer name (String), the customer phone number (String), the customer’s plan choice (String), and monthly bill (double).
   2. A constructor that receives values for the customer name, phone, and plan choice and sets the matching instance variables. It should set the other instance variable to 0.
   3. An abstract void method to compute the monthly bill
   4. An abstract toString method
3. Create a new class named **Customer\_Cable** that is a subclass of **Customer**. It should:
   1. Have a constructor that receives values for the customer name, phone, and plan choice and calls the super class’s constructor sending those values as parameters.
   2. Override the method that computes the bill. If the customer’s plan choice is “Basic”, the monthly bill is 75.00. If it’s “Premium”, the bill is 100.00. If it’s “Platinum”, the bill is 125.00.
   3. Override the toString method. It should return a string saying that this is a cable only customer and have all the instance variables with labels (as usual). Make the bill have currency format.
4. Create a new class named **Customer\_Cable\_Internet** that is a subclass of **Customer**. It should:
   1. Add one more protected String instance variable to hold the Internet speed (“High” or “Regular”)
   2. Override the constructor to receive parameters for name, phone number, plan and Internet speed. It should call the super class constructor sending the first 3 values and then set the Internet speed instance variable equal to the last parameter.
   3. Override the method that computes the bill. If the customer’s plan choice is “Basic”, the monthly bill is 75.00. If it’s “Premium”, the bill is 100.00. If it’s “Platinum”, the bill is 125.00.

If the Internet speed is “High” then add 60.00 to that bill amount. If the speed is “Regular”, then add 40.00 to the bill amount.

* 1. Override the toString method. It should return a string saying that this is a cable and Internet customer and have all the instance variables with labels (as usual). Make the bill have currency format.

1. In the main class (main method) create 2 **Customer\_Cable** objects (2 separate variable names) and read the data for them from the text file (Lab9\_2B.txt).
2. Create 2 **Customer\_Cable\_Internet** objects (2 separate variable names) and read the data for them from the text file too.
3. Print all the objects (using the toString shortcut).

# Part 3

1. Create a new Java project called **Lab9\_2C**
2. Make a secondary class named **Room**. It should have:
   1. 3 private instance variables to hold the room name (String), length (int) and width (int)
   2. A constructor that receives input parameters that will be used to fill in all 3 instance variables.
   3. An int method that returns the room area.
   4. A toString method that returns a string with all the instance variables and the area (with labels)
3. In your main class make an ArrayList of **Room** objects.
4. Read the data from the text file (Lab9\_2C.txt) to create each object and then add it to your ArrayList.
5. Print the ArrayList.
6. Find the Room object in the ArrayList with the smallest area and print the object.