Exercise 1: HelloWorld

On the terminal:

1. create a file named HelloWorld.java
2. In that file, create a class HelloWorld
3. In that class, create a main class method that display “Hello World!”
4. Compile the class HelloWorld using the javac command
5. Run the HelloWorld program using the java command

Hints:

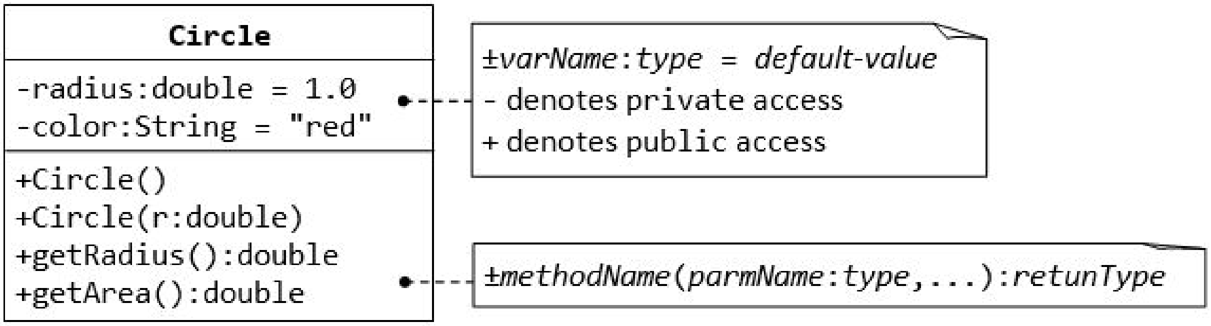
- The signature of the main class method is:
  public static void main(String[] args)
- use System.out.println(String msg) to display on screen the message msg

Exercise 2: HelloWorld Project

1. Run eclipse (from command line, in the Download folder)
2. Create a new project named “HelloWorldProject”
3. Create a class HelloWorld
4. Create a main method that will output “Hello World!”
5. Run the HelloWorld main method

Exercise 3: Circle Class

Create a class Circle such as:



A class called `circle` is designed as shown in the following class diagram. It contains:

- Two `private` instance variables: `radius` (of the type `double`) and `color` (of the type `String`), with default value of `1.0` and `"red"`, respectively.
- Two *overloaded* constructors - a *default* constructor with no argument, and a constructor which takes a double argument for radius.
- Two `public` methods: `getRadius()` and `getArea()`, which return the radius and area of this instance, respectively.

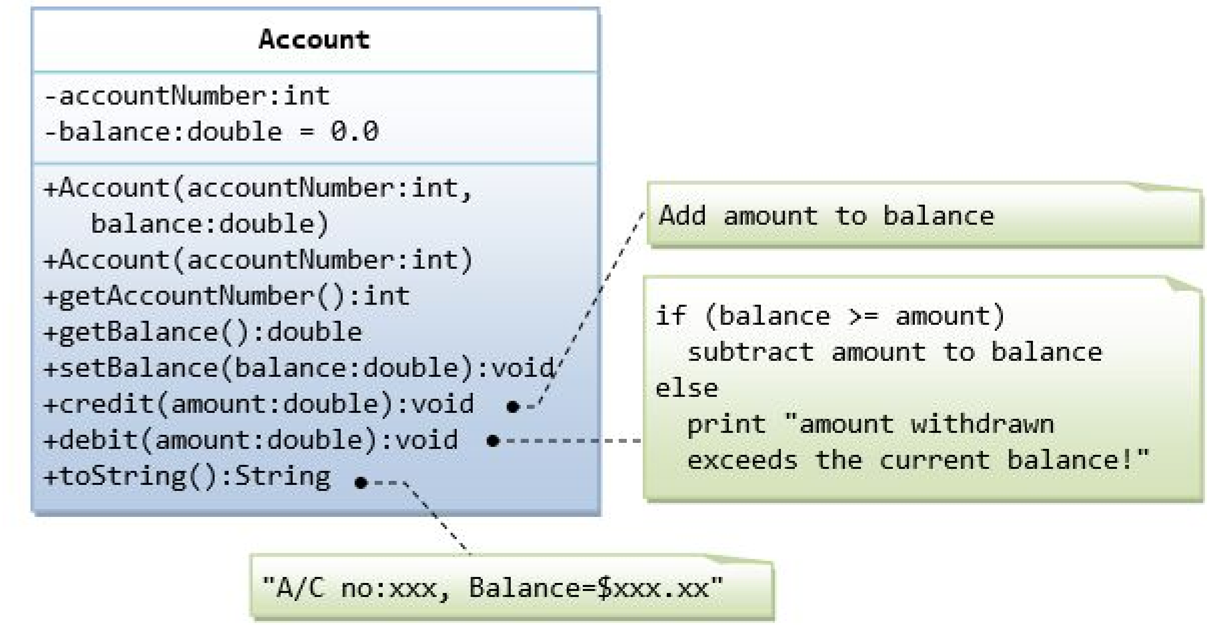
A Test Driver for the `Circle` Class

```
// A Test Driver for the Circle class
public class TestCircle {
   public static void main(String[] args) {
      // Test constructors and toString()
      Circle c1 = new Circle(1.1, "blue");
      System.out.println(c1);  // toString()
      Circle c2 = new Circle(2.2);
      System.out.println(c2);  // toString()
      Circle c3 = new Circle();
      System.out.println(c3);  // toString()

      // Test Setters and Getters
      c1.setRadius(2.2);
      c1.setColor("green");
      System.out.println(c1);  // toString() to inspect the modified instance
      System.out.println("The radius is: " + c1.getRadius());
      System.out.println("The color is: " + c1.getColor());

      // Test getArea() and getCircumference()
      System.out.printf("The area is: %.2f%n", c1.getArea());
      System.out.printf("The circumference is: %.2f%n", c1.getCircumference());
   }
}
```

The expected outputs are:

```
Circle[radius=1.1, color=blue]
Circle[radius=2.2, color=red]
Circle[radius=8.8, color=red]
Circle[radius=2.2, color=green]
Radius is: 2.2
Color is: green
Area is: 15.21
Circumference is: 13.82
```

Exercise 4: Account Class


Account
-accountNumber:int
-balance:double = 0.0
+Account(accountNumber:int,
balance:double)
+Account(accountNumber:int)
+getAccountNumber():int
+getBalance():double
+setBalance(balance:double):void
+credit(amount:double):void
+debit(amount:double):void
+toString():String
Add amount to balance
if (balance >= amount)
subtract amount to balance
else
print "amount withdrawn exceeds the current balance!"
"A/C no:xxx, Balance=$xxx.xx"

A class called `Account`, which models a bank account, is designed as shown in the class diagram. It contains the following members:

- Two `private` instance variables: `accountNumber` (`int`), and `balance` (`double`) which maintains the current account balance.
- Constructors (overloaded).
- Getters and Setters for the `private` instance variables. There is no setter for `accountNumber` as it is not designed to be changed.
- `public` methods `credit()` and `debit()`, which adds/subtracts the given `amount` to/from the balance, respectively.
- A `toString()`, which returns "`A/C no:xxx, Balance=$xxx.xx`", with `balance` rounded to two decimal places.

Write the `Account` class and a test driver to test all the `public` methods.

Hint: see `String.format options for the toString method`

A Test Driver for the `Account` Class (`TestAccount.java`)

```
/*
 * A Test Driver for the Account class.
 */
public class TestAccount {
   public static void main(String[] args) {
      // Test Constructors and toString()
      Account a1 = new Account(1234, 99.99);
      System.out.println(a1);  // toString()
      Account a2 = new Account(8888);
      System.out.println(a2);  // toString()

      // Test Setters and Getters
      a1.setBalance(88.88);
      System.out.println(a1);  // run toString() to inspect the modified instance
      System.out.println("The account Number is: " + a1.getAccountNumber());
      System.out.println("The balance is: " + a1.getBalance());

      // Test credit() and debit()
      a1.credit(10);
      System.out.println(a1);  // run toString() to inspect the modified instance
      a1.debit(5);
      System.out.println(a1);
      a1.debit(500);   // Test debit() error
      System.out.println(a1);
    }
}
```

The expected outputs are:

```
A/C no:1234, Balance=99.99
A/C no:8888, Balance=0.00
A/C no:1234, Balance=88.88
Account Number is: 1234
Balance is: 88.88
A/C no:1234, Balance=98.88
A/C no:1234, Balance=93.88
amount withdrawn exceeds the current balance!
A/C no:1234, Balance=93.88
```