# Object-Oriented Programming Fundamentals

## Lecture/Workshop (Week 6)

**Defining Classes, Creating Objects, toString()**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Task 1**

Remember that generally, the steps to define a class are

1. Sketch a model of the class
2. Define the class header
3. Define the attributes
4. Define the constructors
5. Define the methods

Define a **Circle** class that

* Has a **radius**
* Allows the user to
* display the radius
* calculate and display the diameter
* calculate and display the perimeter
* calculate and display the area
* display all the measurements at once

**Example – travel money card**

Create and test a class representing a travel money card that

* Allows a traveller to load money of a particular currency onto a card and access it overseas
* Has a card number, a name, a currency and a balance
* Allows a user to add funds, make purchases, and check the balance

Rules

* Card numbers and names must not be missing
* Balance must not be negative

**Defining class MoneyCard**

To define the **MoneyCard** class and test it, we go through the following steps

* Sketch a model of the class
* Define the class header
* Define the attributes
* Define the constructors
* Define the methods

**Sketch a model of the class**

|  |
| --- |
| **MoneyCard** |
| private String cardNumber  private String customerName  private String currency  private double balance |
| public MoneyCard(String cardNo, String custName, String currType)  public void addFunds(double amount)  public void makePurchase(double amount)  public double getBalance()  public void displayMoneyCard()  public String toString() |

**Method to display a MoneyCard object**

**public void displayMoneyCard()**

**{**

**System.out.println("MoneyCard details: "**

**+ "\n cardNumber: " + cardNumber**

**+ "\n customerName: " + customerName**

**+ "\n currency: " + currency**

**+ "\n balance: " + balance);**

**}**

**toString() method**

* Often we wish to display all the details of an object to the screen
* One approach is a display method like **displayMoneyCard()** above
* Sometimes we want to send the details to the screen or alternatively to somewhere else such as a file
* The **toString()** method is a special method that returns the content of the object as a **String** variable

**public String toString()**

**{**

String description = "MoneyCard["

+ " cardNumber: " + cardNumber

+ " customerName: " + customerName

+ " currency: " + currency

+ " balance: " + balance + "]";

return description;

**}**

**Testing the MoneyCard class**

* Define a launcher class to test **MoneyCard** (this class has a **main()** method)
* We create instances of the **MoneyCard** class and send messages to them

**Testing**

**// Test 1 - create a money card and display it**

MoneyCard a1 = new MoneyCard("C10", "Smith", "EUR");

System.out.println(a1.toString());

**// Test 2 – add funds and display the money card**

a1.addFunds(200);

System.out.println(a1);

**// Test 3 – try to add an invalid amount and see**

**// how the object handles it**

a1.addFunds(-100);

System.out.println(a1);

**// Test 4 - make a purchase**

a1.makePurchase(100);

System.out.println(a1);

**// Test 5 - make an invalid purchase request**

a1.makePurchase(0);

System.out.println(a1);

**// Test 6 - make a purchase request that is too large**

a1.makePurchase(300);

System.out.println(a1);

**// Test 7 - get the balance**

double balance = a1.getBalance();

System.out.println("balance: " + balance);

**// Test 8 – display the money card details**

a1.displayMoneyCard();

**----------------------------------------------------------------**

**Wanting help?** Join us at the Code Hub, session times listed in the ‘Need Help’ section of LMS