ADVANCED OBJECT ORIENTED PROGRAMMING

Engr. Anees Ahmed Soomro

Assistant Professor

CS QUEST Nawabshah

https://anees-soomro.neocities.org

OOP CONCEPTS

Objects , Object references, Messages.

Declaring Members: Fields and Methods

- Here is the class **Check**
- A class definition consists of a series of member declarations. In the case of the class **Check**, it has one field:
- **amount**, which is an integer to hold value of Check.

- The class Check has two methods that implement the essential operations on a Check:
- **setAmount**(int value) : To adjust the value of check.
- **getAmount**(): It retrieves and return the amount which is adjusted by setAmount() method.

  - The class definition also has a method-like declaration with the same name as the class.
  - Such declarations are called constructors. As we shall see, a constructor is executed when an object is created from class.
  - However, the implementation details in the example are not important for the present discussion.

```
class Check
{
private int amount=0;
    public int getAmount()
{     return amount; }
    public void setAmount(int amt)
{     amount=amt;}
}
public class Mainclass
{
    public static void main(String[] args)
{
        int amt=0;
        Check obj= new Check();
        obj.setAmount(200);
        amt=obj.getAmount();
        System.out.println("Your current amount is :"+amt);
        }
   }
```

(OBJECTS):-Class Instantiation

- The process of creating objects from a class is called **instantiation**. An object is an instance of a class.
- The object is constructed using the class as a blueprint and is a concrete instance of the abstraction that the class represents.
- An object must be created before it can be used in a program.
- In Java, objects are manipulated through object references (also called reference values or simply references).
- The process of creating objects usually involves the following steps:

Declaration of a variable to store the object reference

- This involves declaring a reference variable of the appropriate class to store the reference to the object.

// Declaration of two reference variables that will denote

// Two distinct objects, namely two checks of characters, respectively.

Check ch1, ch2;

Creating an object

- This involves using the new operator in conjunction with a call to a constructor, to create an instance of the class.
- // Create two distinct stacks of chars.

ch1 = new Check(); // Check amount

ch2 = new Check(); // Check amount:

The new operator returns a reference to a new instance of the **Check** class.

- This reference can be assigned to a reference variable of the appropriate class.
- Each object has a unique identity and has its own copy of the fields declared in the class definition.
- The two checks, denoted by ch1 and ch2, will have their own **amount** field.
- The purpose of the constructor call on the right side of the new operator is to initialize the newly created object.
- In this particular case, for each new Check instance created using the new operator, the constructor creates check for amount setting and retrieval.

- The declaration and the instantiation can also be combined:

```
Chek ch1 = new Check(), ch2 = new Check();
```

- If the name of the reference variable is omitted, this denotes an anonymous object.
- Since objects in Java do not have names, but are denoted by references.

Object References

- A reference provides a handle to an object that is created and stored in memory.
- Objects can only be manipulated via references, which can be stored in variables.
- An object can have several references, often called its **aliases**.
- The object can be manipulated via any one of its **aliases**.
- // Create two distinct checks.

Check checkA = new Check();

Check checkB = new Check();

checkB = checkA;// (1) aliases after assignment//

Check previously referenced by checkB can now be garbage collected.

- Two checks are created in the code. Before assignment, After assignment at (1), reference variables checkA, checkB will denote same Check,
- Reference variables checkA and checkB are aliases after the assignment, as they refer to the same object.
- What happens to the check object that was denoted by the reference variable checkB before the assignment?
- When objects are no longer in use, their memory is, if necessary, reclaimed and reallocated for other objects.
- This is called automatic garbage collection. Garbage collection in Java is taken care of by the runtime system.

Messages

- **Message passing** is a form of communication used in object oriented programming.
- Communication is made by the sending of messages to recipients.
- Forms of messages include function invocation, signals, and data packets.
- The process by which an object sends data to another object or asks the other object to invoke a method.
- Also known to some programming languages as **interfacing**.
- For example, the object called Breeder may tell the Lassie object to sit by passing a "sit" message which invokes Lassie's "sit" method.
- The syntax varies between languages, for example: [Lassie sit] in Objective-C.
- In Java code-level message passing corresponds to "method calling".
- Some dynamic languages use double-dispatch or multi-dispatch to find and pass messages.
- Message passing systems have been called "**shared nothing**" systems because the message passing abstraction hides underlying state changes that may be used in the implementation of sending messages.
- In the terminology of OOP languages, a message is single means to pass control to an object.
- If the object 'responds' to the message, it has a method for that message.