Project 2-Individual assignment

Mini team sports game simulation (80 points)

Due: 12/07 Wednesday 11:59 pm

No late submission accepted

Computer Gaming Simulation

Computer gaming is a computer simulation of a virtual world

Game designers must have knowledge of the following to make people, objects, and environments behave realistically in a virtual world:

- Computer graphics
- Artificial intelligence
- Human-computer interactions and simulation
- Software engineering
- Computer security
- Fundamentals of mathematics
- Laws of physics relating to gravity, elasticity, light, and sound

https://www.youtube.com/watch?v=I6AjwpwxhoQ

Creating the Virtual World

Game engine--a software system within which games can be created

Following functionality provided by tools of a game engine:

- A rendering engine for graphics
- A physics engine to provide a collision detection system and dynamics simulation
- A sound-generating component

Creating the Virtual World

Additional functionality resulting from tools of a game engine:

- A scripting language apart from the code driving the game
- Animation
- Artificial intelligence algorithms (e.g., path-finding algorithms)
- A scene graph that holds the spatial representation in a graphical sense

Soft Skills

High Quality Game Design and Development Requires Effective Use of “Soft Skills”:

- Effective collaboration with designers, programmers, and artists on various technical ideas throughout the entire game design and development process
- Flexibility and adaptability as the game design constantly evolves and changes throughout the development and production process

Soft Skills

- Willingness to abandon much of the completed design work when the game's story line, mechanics, art, programming, audio, video, and/or scripting requires significant changes

Game Programming

A variety of application programming interfaces (APIs) and libraries are available to help developers with key programming tasks

The choice of API determines which vocabulary and calling conventions the programmer should employ to use the services

The target game platform determines which service the programmer will use; some libraries permit efficient cross-platform development

Game Programming

Coding process begins with the creation of “the game loop”

Game loop is responsible for managing the game world, regardless of any input from the user

For example, the game loop might update enemy movement in the game or check for victory/loss conditions

Basically, the game loop manages the simulation

What Is Simulation?

Simulation

A model of a complex system and the experimental manipulation of the model to observe the results

- Systems that are best suited to being simulated are **dynamic**, **interactive**, and **complicated**

Model

An abstraction of a real system

- It is a representation of the objects within the system and the rules that govern the interactions of the objects

Constructing Models

Continuous simulation

- Treats time as continuous
- Expresses changes in terms of a set of differential equations that reflect the relationships among the set of characteristics
- Meteorological models fall into this category

Thinking Machines

https://www.youtube.com/watch?v=QdQL11uWWcI

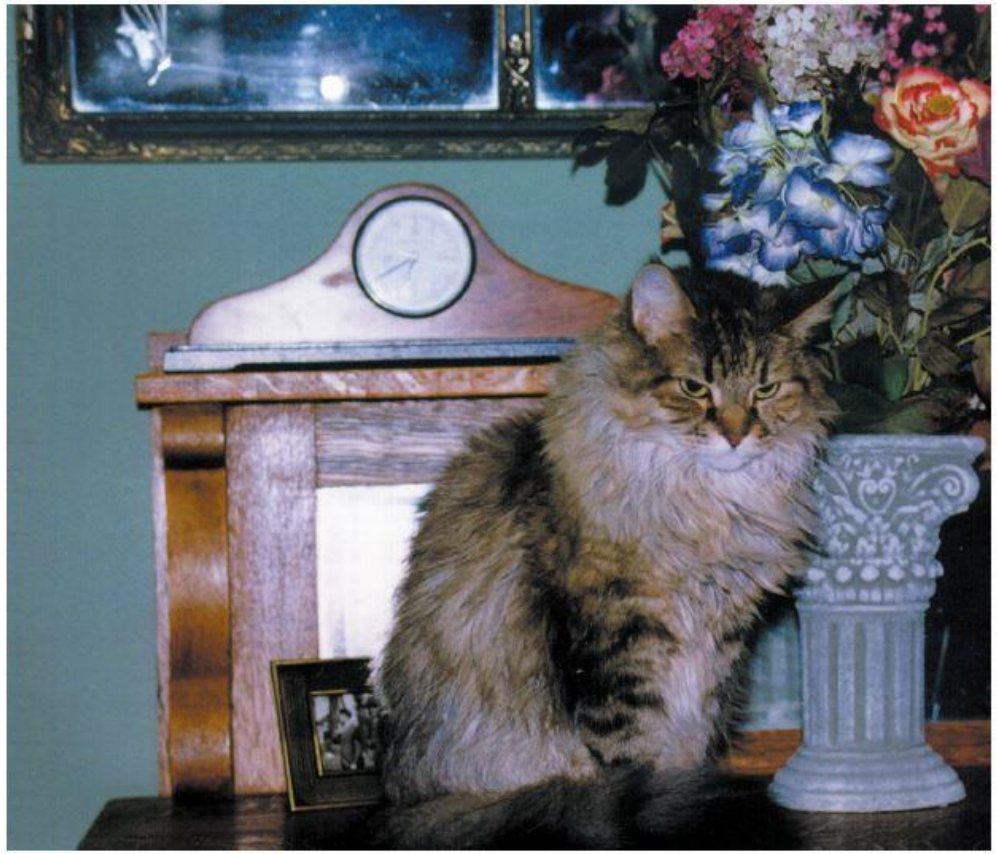


FIGURE 13.1 A computer might have trouble identifying the cat in this picture

Courtesy of Amy Rose

http://www.theverge.com/2013/9/17/4740466/how-siri-found-its-voice-video

Can you list the items in this picture?

Thinking Machines

Humans do best

Can you list the items in this picture?

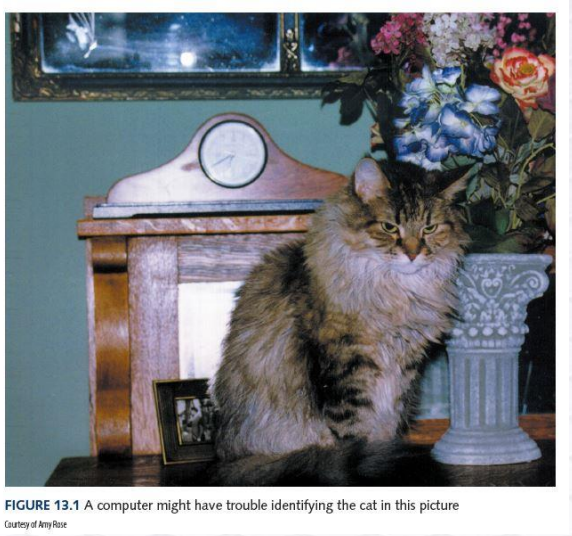


FIGURE 13.1 A computer might have trouble identifying the cat in this picture

Courtesy of Amy Rose

Computers do best

Can you count the distribution of letters in a book?

Add a thousand4-digit numbers? Match finger prints?

Search a list of a million values for duplicates?

Thinking Machines

Artificial intelligence (AI)

The study of computer systems that attempt to model and apply the intelligence of the human mind

For example, writing a program to pick out objects in a picture

The Turing Test

Turing test

A test to empirically determine whether a computer has achieved intelligence

Alan Turing

An English mathematician who wrote a landmark paper in 1950 that asked the question: *Can machines think?*

He proposed a test to answer the question "How will we know when we've succeeded?"

The Turing Test

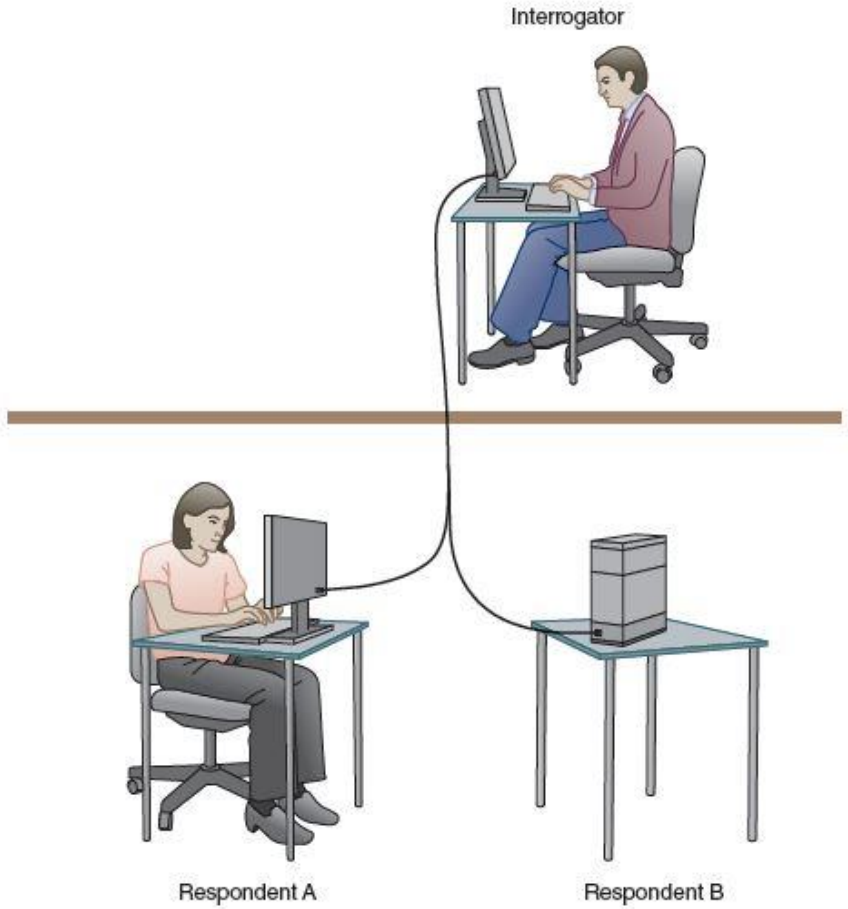
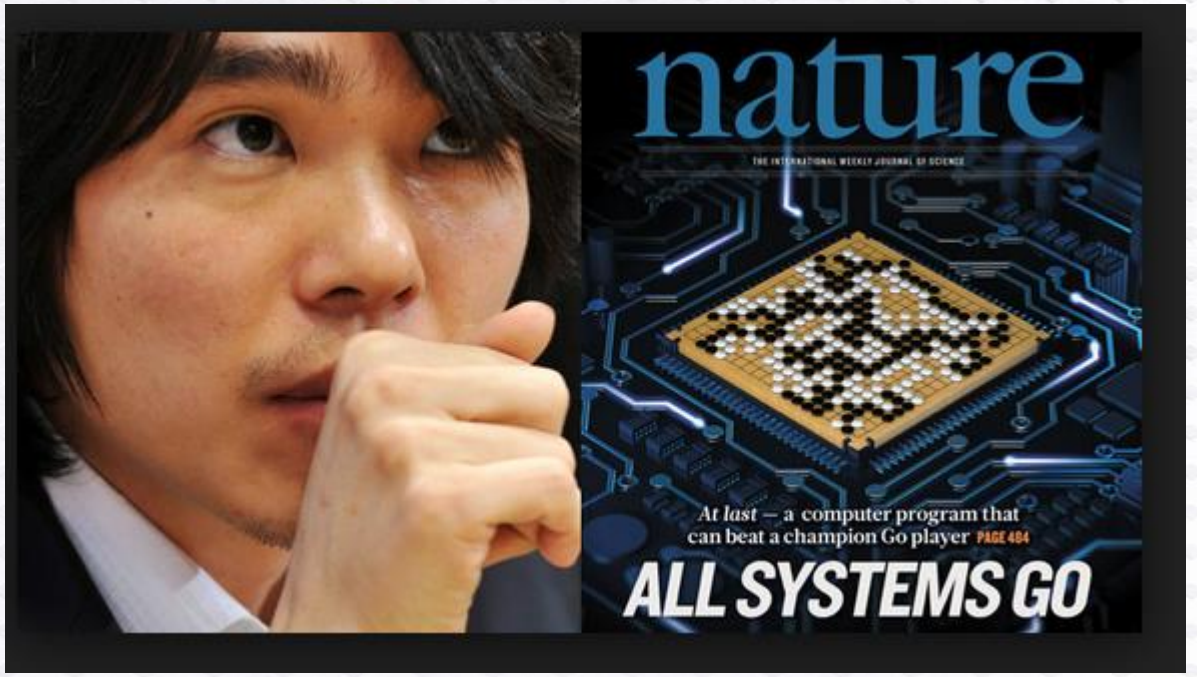
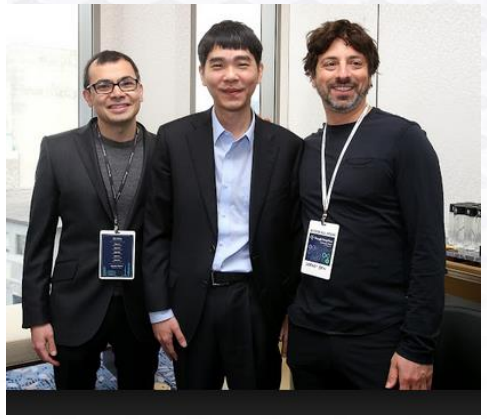


FIGURE 13.2 In a Turing test, the interrogator must determine which respondent is the computer and which is the human





https://deepmind.com/alpha-go.html

Google DeepMind

The Turing Test

Weak equivalence

Two systems (human and computer) are equivalent in results (output), but they do not arrive at those results in the same way

Strong equivalence

Two systems (human and computer) use the same internal processes to produce results

Robotics

Mobile robotics

The study of robots that move relative to their environment, while exhibiting a degree of autonomy

Sense-plan-act (SPA) paradigm

The world of the robot is represented in a complex semantic net in which the sensors on the robot are used to capture the data to build up the net

FIGURE 13.8 The sense–plan–act (SPA) paradigm

Subsumption Architecture

Rather than trying to model the entire world all the time, the robot is given a simple set of behaviors each associated with the part of the world necessary for that behavior

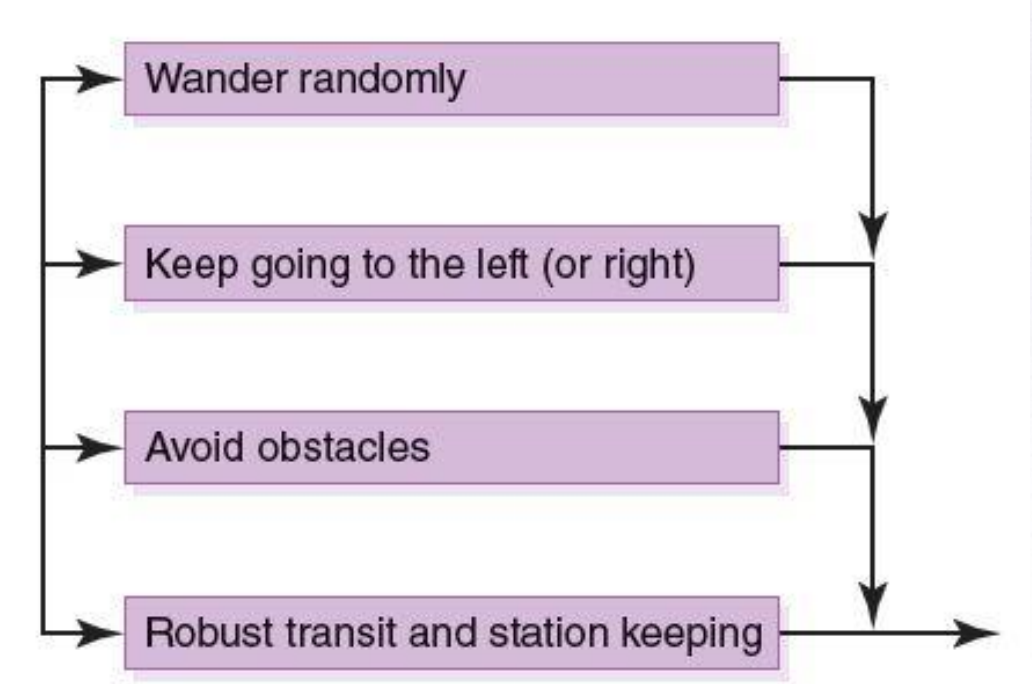
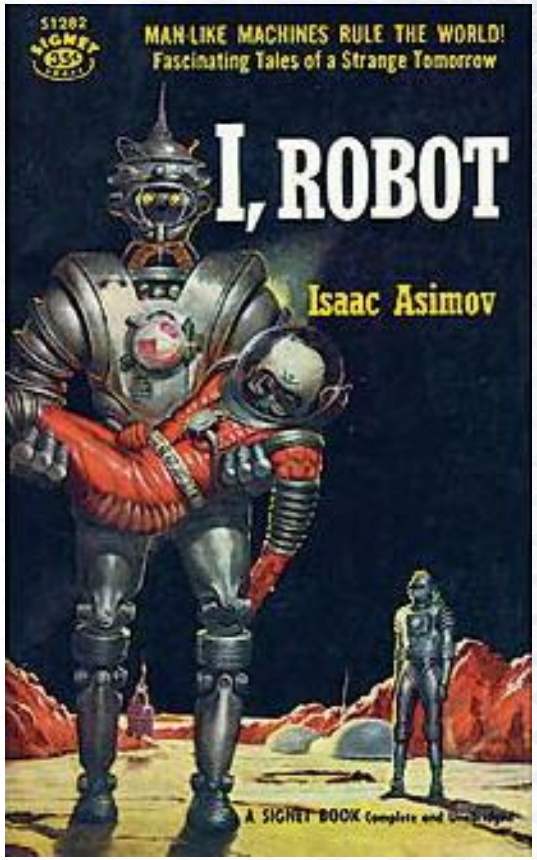


FIGURE 13.9 The new control paradigm

Subsumption Architecture



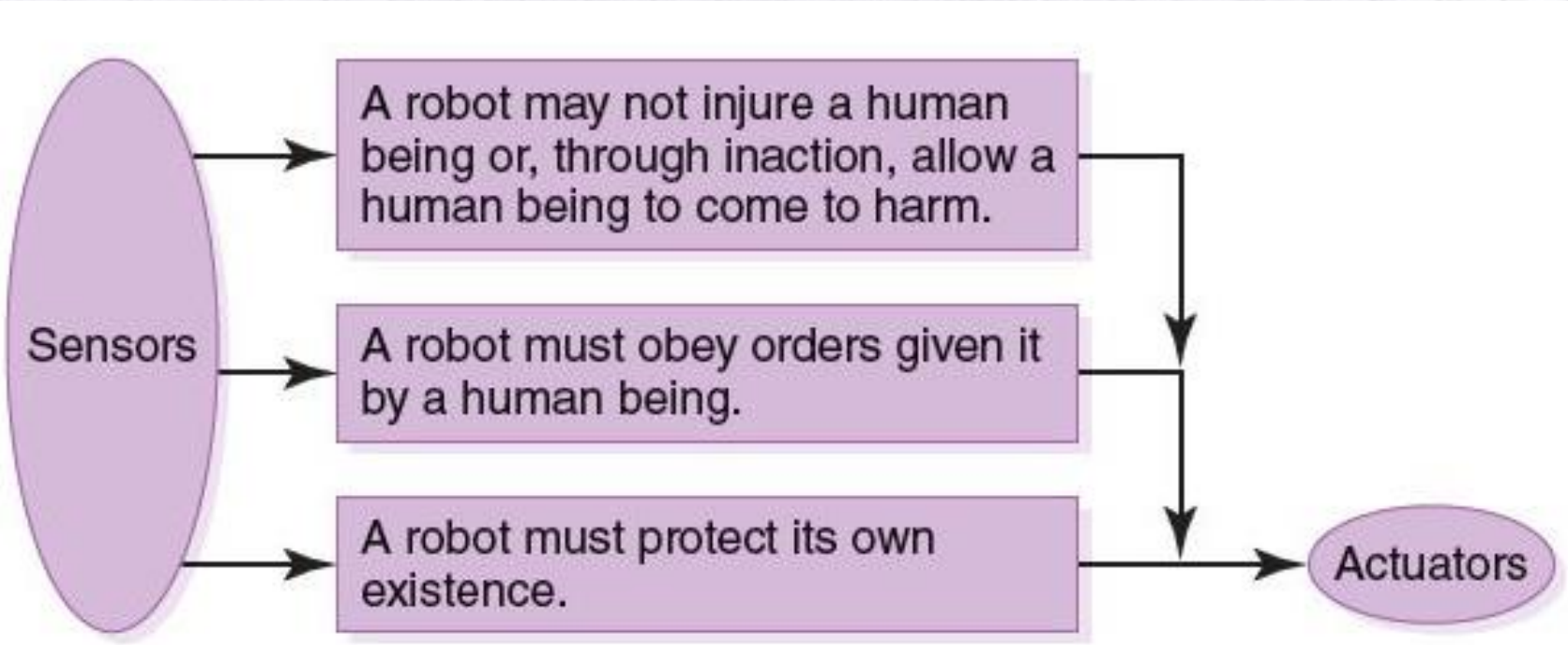


FIGURE 13.10 Asimov's laws of robotics are ordered

Recursive Searching: Simple maze!

4 4 4 4 4 4 4
4 0 0 0 0 0 4
4 0 4 0 4 0 4
4 0 0 4 0 4 4
4 0 4 0 0 0 4
4 0 0 0 4 0 4
4 4 4 4 4 4 4

4 4 4 4 4 4 4
4 7 7 7 7 7 4
4 7 4 7 4 7 4
4 7 7 4 0 4 4
4 7 4 7 7 7 4
4 7 7 7 4 7 4
4 4 4 4 4 4 4

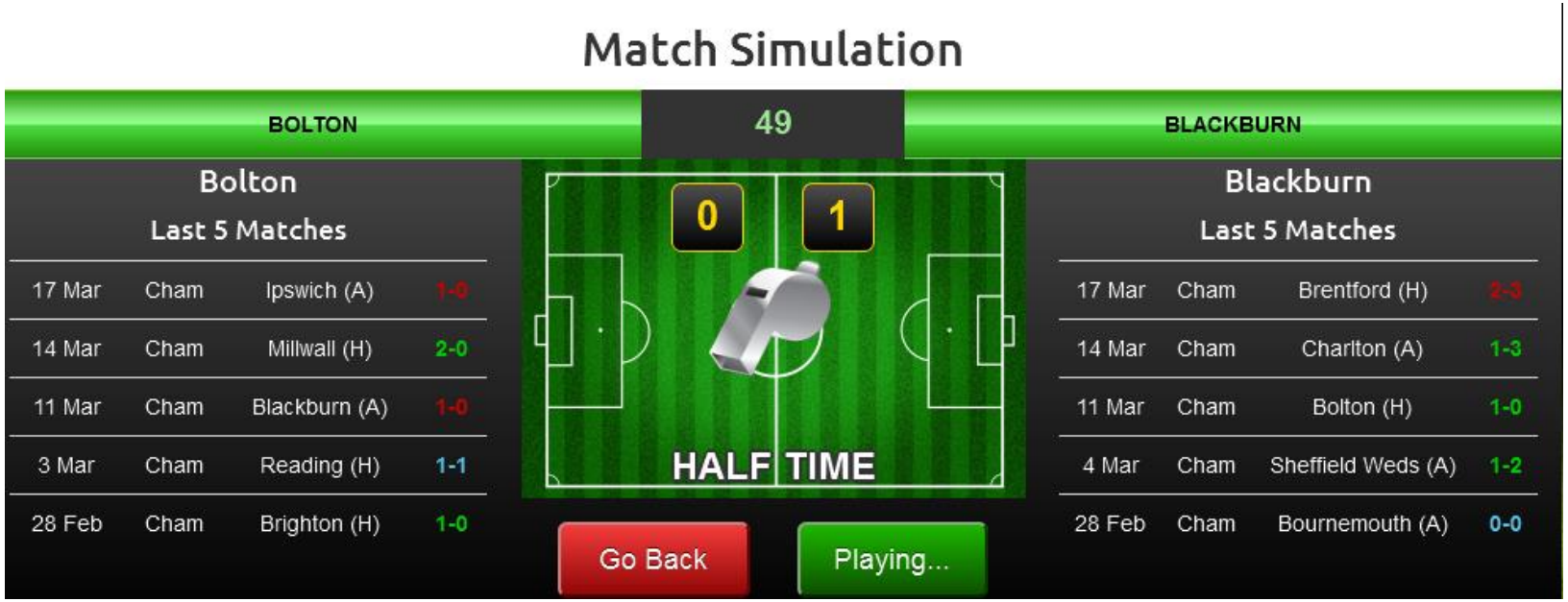
4 4 4 4 4 4 4
4 7 7 7 7 7 4
4 7 4 7 4 7 4
4 7 7 4 0 4 4
4 7 4 7 7 7 4
4 7 7 7 4 7 4
4 4 4 4 4 4 4

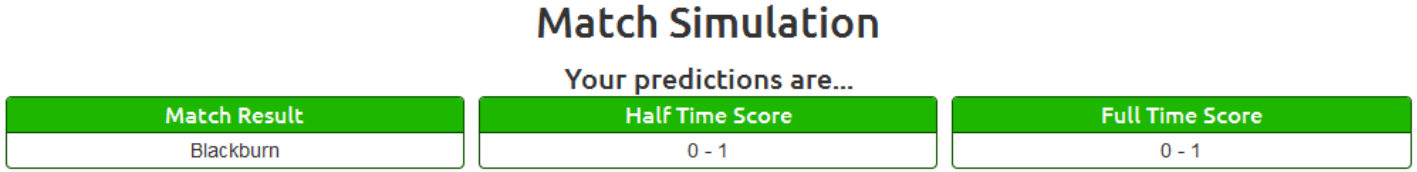
Project 2

Mini team sports game simulation (80 points)

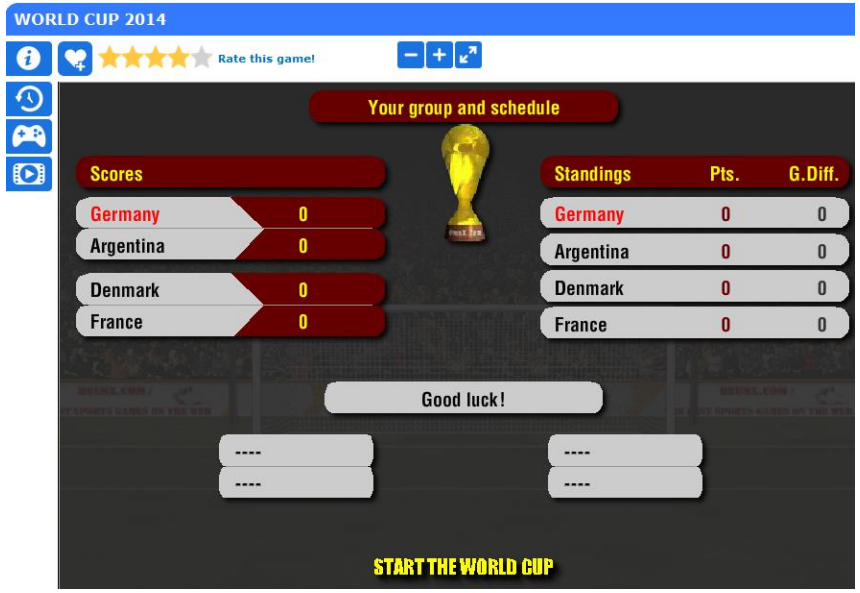
Due: 8/21 Sunday 11:59 pm
No late submission accepted

- **Soccer Rules: The basic rules of soccer for kids and adults.**
  - https://www.youtube.com/watch?v=M18UNJDspO4
- **Football Simulator**
  - http://www.betstatz.com/simulator





http://www.agame.com/games/soccer



http://www.y8.com/tags/soccer

http://www.y8.com/games/the_champions_4_-_world_domination

http://www.easports.com/2014-fifa-world-cup/news/2014/ea-sports-2014-fifa-world-cup-prediction

http://keithlyons.me/blog/2014/05/30/predicting-the-outcome-of-the-2014-fifa-world-cup/

http://www.ea.com/uk/news/national-team-infographics

The World Football Elo Rating System

The ratings are based on the following formulas:

$$R_n = R_o + K \times (W - W_e)$$

R_n is the new rating, R_o is the old (pre-match) rating.

K is the weight constant for the tournament played:

- **60** for World Cup finals;
- **50** for continental championship finals and major intercontinental tournaments;
- **40** for World Cup and continental qualifiers and major tournaments;
- **30** for all other tournaments;
- **20** for friendly matches.

K is then adjusted for the goal difference in the game. It is increased by **half** if a game is won by two goals, by **3/4** if a game is won by three goals, and by **3/4 + (N-3)/8** if the game is won by four or more goals, where **N** is the goal difference.

W is the result of the game (**1** for a win, **0.5** for a draw, and **0** for a loss).

W_e is the expected result (win expectancy), either from the chart or the following formula:

$$W_e = 1 \ / \ (10^{(-dr/400)} + 1)$$

dr equals the difference in ratings plus **100** points for a team playing at home.

http://www.eloratings.net/system.html

Video game simulates 2014 Stanley Cup final

http://www.sportsnet.ca/hockey/nhl/ea-sports-nhl-14-predicts-boston-bruins-stanley-cup-san-jose-sharks/

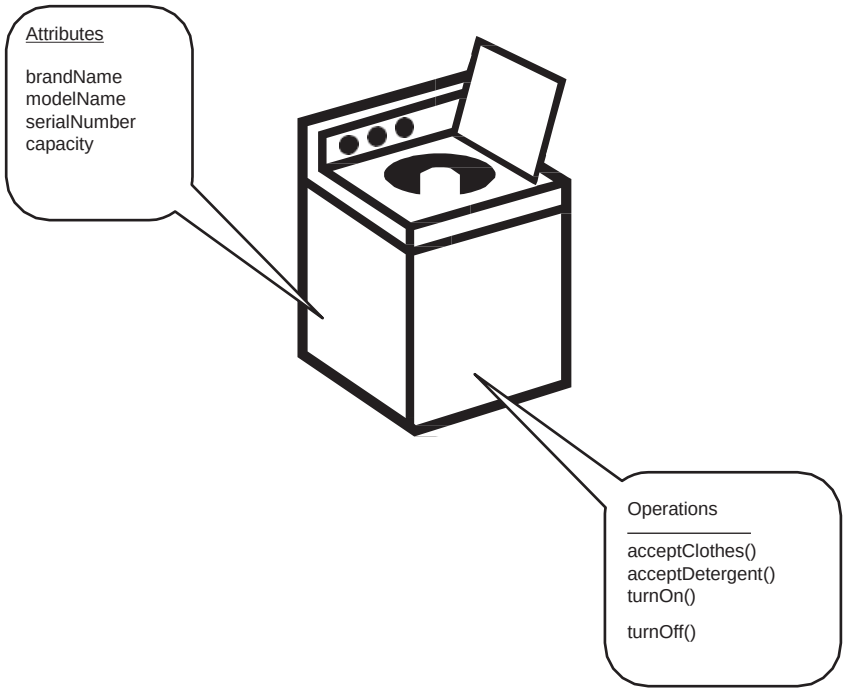
Project Requirement
1. UML Diagram document

Unified Modeling Language (UML)

- The UML is a visual modeling language that enables system builders to create blueprints that capture their visions in a standard, easy-to-understand way, and provides a mechanism to effectively share and communicate these visions with others.

The key is to organize the design process in a way that analysts, clients, programmers, and others involved in system development can understand and agree on.
The UML provides the organization.

WashingMachine class is a template for creating new instances of washing machines.

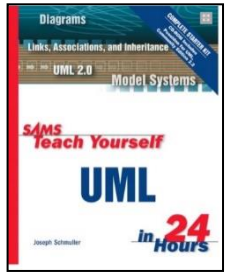


Class Diagram

WashingMachine
brandName modelName serialNumber capacity
acceptClothes() acceptDetergent() turnOn() turnOff()

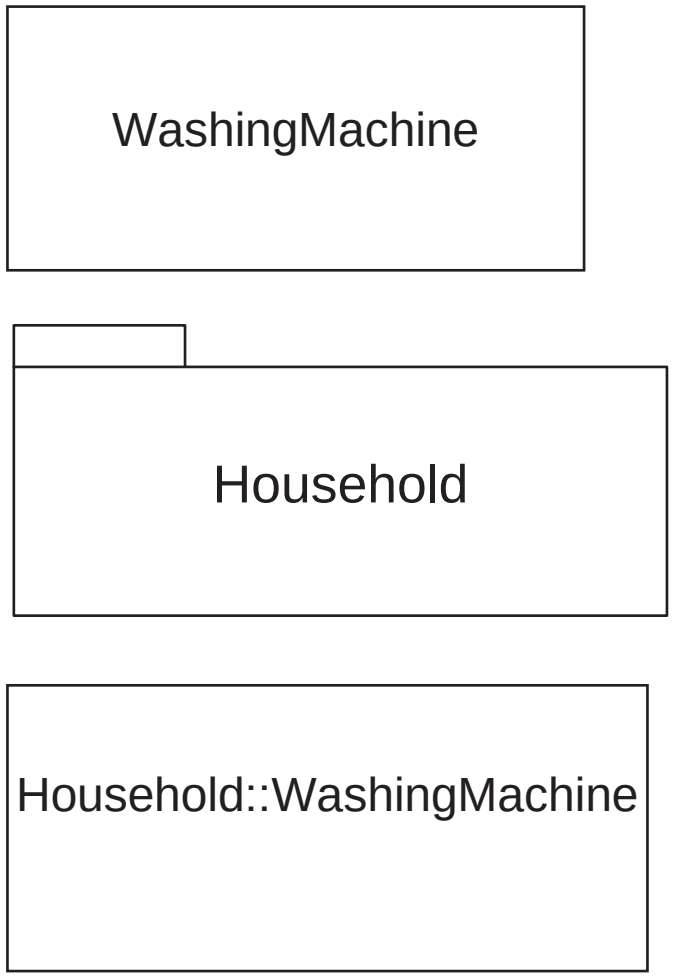
Object Diagram

myWasher:WashingMachine	:WashingMachine



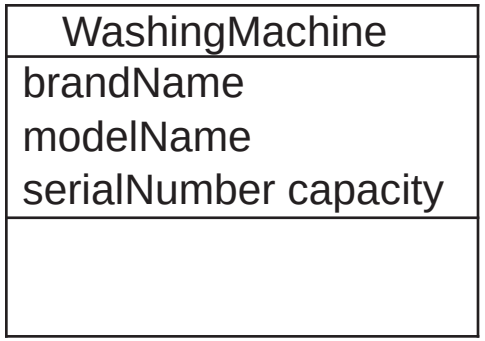
Working with Object- Orientation

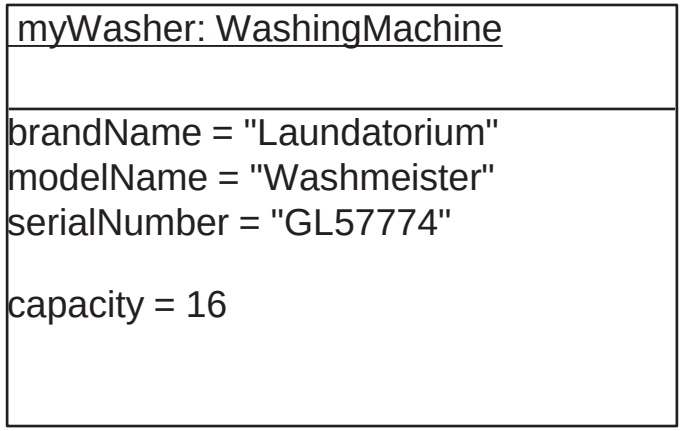
Visualizing a Class



- a rectangle is the icon that represents a class
- If your class has a two- word name, join the two words together and capitalize the first letter of the second word (as in WashingMachine)
- the UML represents a package as a tabbed folder.
- If the WashingMachine class is part of a package called Household, you can give it the name Household::WashingMachine. The double colons separate the package name on the left from the classname on the right. This type of classname is called a pathname

Attributes

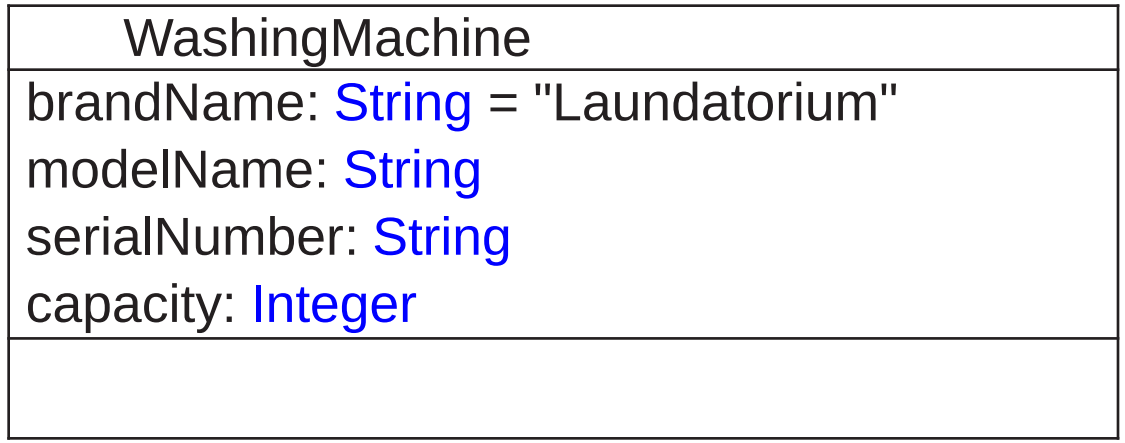




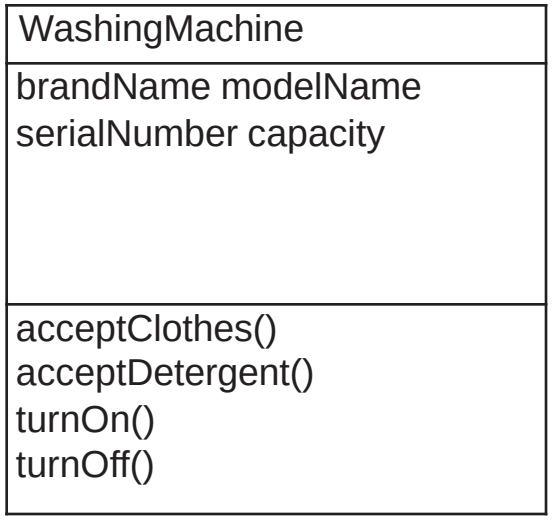
An object has a specific value for every one of its class's attributes.

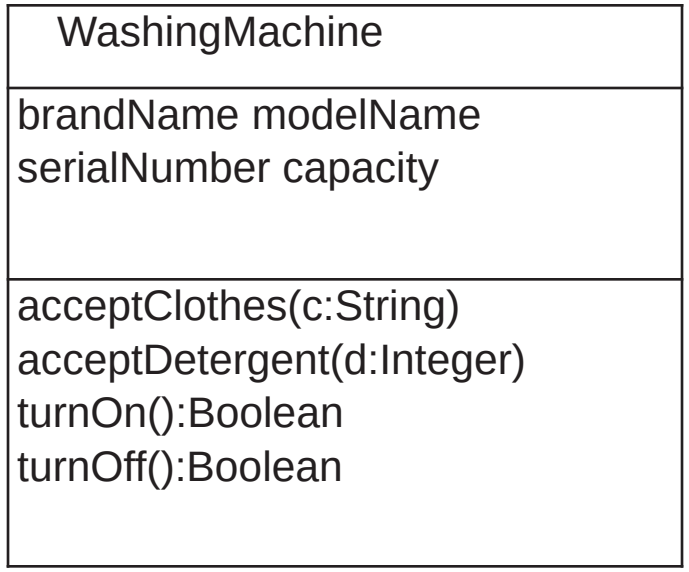
- An attribute is a property of a class. It describes a range of values that the property may hold in objects (that is, in instances) of that class. A class may have zero or more attributes.
- By convention, a one-word attribute name is written in lower- case letters. If the name consists of more than one word, the words are joined and each word other than the first word begins with an uppercase letter. The list of attribute names begins below a line separating them from the class name.
- Every object of the class has a specific value for every attribute. Note that an object's name begins with a lowercase letter, pre-cedes a colon that precedes the class name, and the whole name is underlined.

An attribute can show its type as well as a default value.



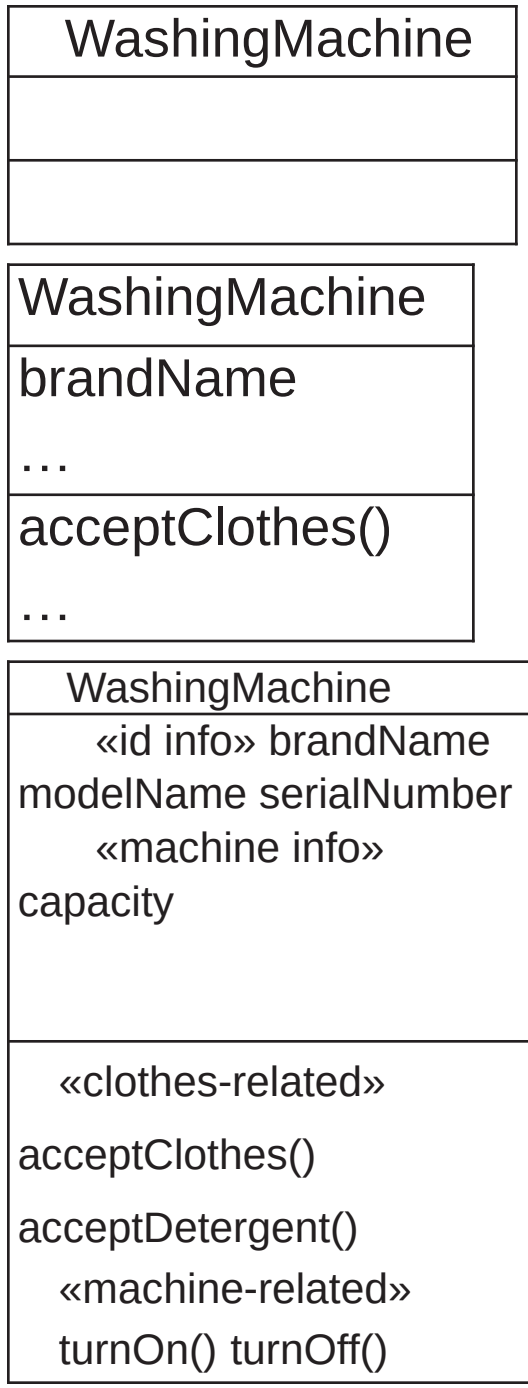
- The UML gives you the option of indicating additional information for attributes. In the icon for the class, you can specify a type for each attribute's value. Possible types include string, floating-point number, integer, and Boolean (and other enu- merated types). To indicate a type, use a colon to separate the attribute name from the type. You can also indicate a default value for an attribute. a





Operations

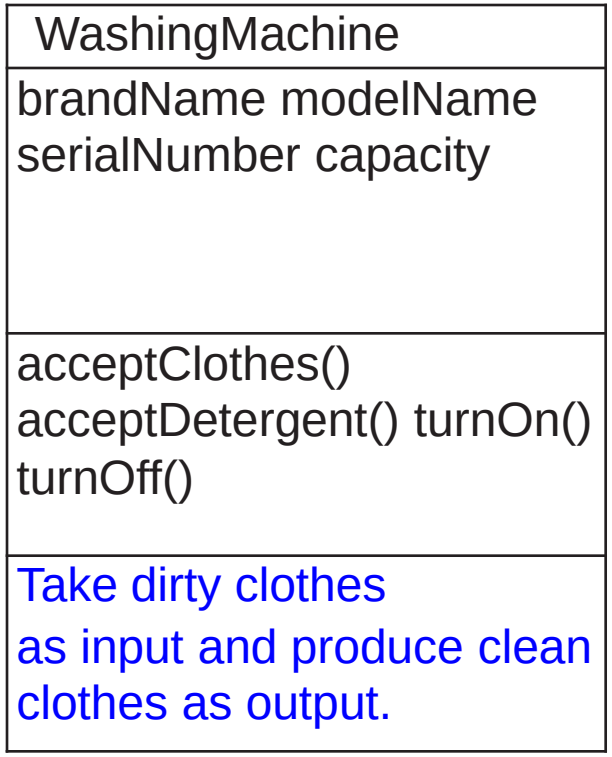
- An operation is something a class can do, and hence it is something that you (or another class) can ask the class to do.
- The list of operations begins below a line that separates the operations from the attributes
- In the parentheses that follow an operation name, you can show the parameter that the operation works on, along with that parameter's type. One kind of operation, the function, returns a value after it finishes doing its work. For a function, you can show the value it returns and that value's type.
- These pieces of information about an operation are called the operation's signature. The first two operations show the type of the parameter. The third and fourth show the type of the return value.



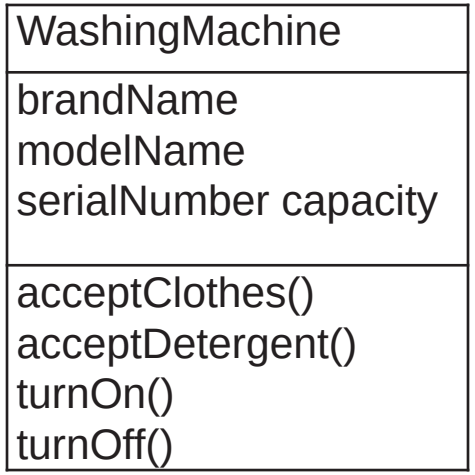
Attributes, Operations, and Visualization

- In practice, you don't always show all of a class's attributes and operations
- An ellipsis indicates that the displayed attributes or operations aren't the whole set.
- You can use a keyword to organize a list of attributes or operations.

Responsibilities and Constraints



- In a class icon, you can write the class's responsibilities in an area below the operations list area.

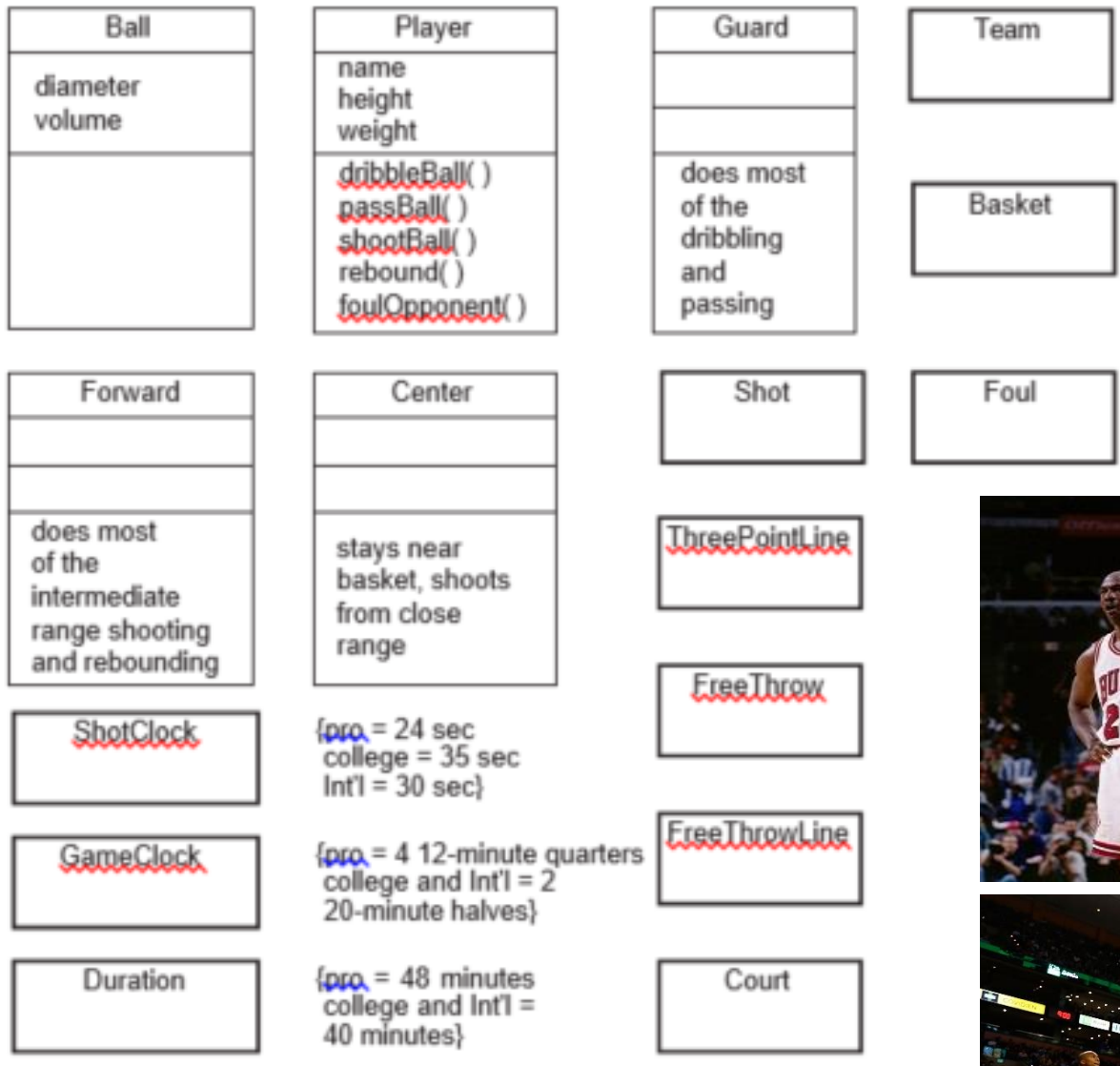


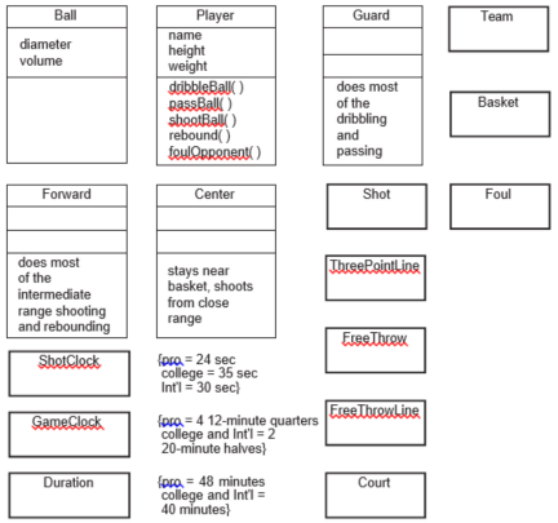
- The rule in curly brackets constrains the capacity attribute to be one of three possible values.

{capacity = 16 or 18 or 20 lb}

Classes—What They Do and How to Find Them

- An initial class diagram for modeling the game of basketball.





Analyst: "Coach, what's basketball all about?"

Coach: "The goal of the game is to shoot the ball through the basket and score more points than your opponent. Each team consists of five players: two guards, two forwards, and a center. Each team advances the ball toward the basket with the objective of ultimately shooting the ball through the basket."

Analyst: "How does it advance the ball?"

Coach: "By dribbling and passing. But the team has to take a shot at the basket before the shot clock expires."

Analyst: "Shot clock?"

Coach: "Yes. That's 24 seconds in the pros, 30 seconds in international play, and 35 seconds in college to take a shot after a team gets possession of the ball."

Analyst: "How does the scoring work?"

Coach: "Each basket counts two points, unless the shot is from behind the three-point line. In that case, it's three points. A free throw counts one point. A free throw, by the way, is the penalty a team pays for committing a foul. If a player fouls an opponent, play stops and the opponent gets to shoot at the basket from the free-throw line."

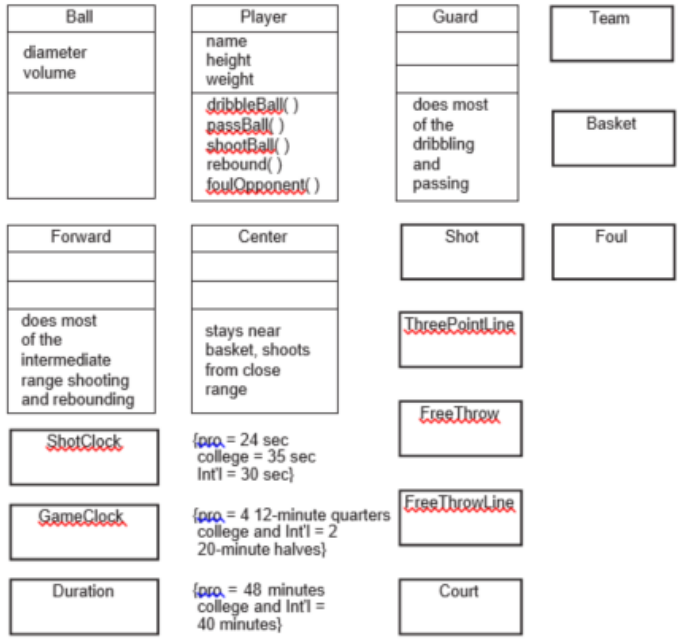
Analyst: "Tell me a little more about what each player does."

Coach: "The guards generally do most of the dribbling and passing. They're typi-cally shorter than the forwards, and the forwards are usually shorter than the center. All the players are supposed to be able to dribble, pass, shoot, and rebound. The forwards do most of the rebounding and intermediate-range shoot- ing, while the center stays near the basket and shoots from close range."

Analyst: "How about the dimensions of the court? And by the way, how long does a game last?"

Coach: "In international play, the court is 28 meters long by 15 meters wide. The basket is 10 feet off the ground. In the pros, a game lasts 48 minutes, divided into four 12-minute quarters. In college and international play, it's 40 minutes divided into two 20-minute halves. A game clock keeps track of the time remaining."

UML Project Diagram



```cpp
#include <iostream>
#include <string>
using namespace std;

class Students {

public:
    Students();
    Students(string nn, int aa, int ii)
    {
        name = nn;
        age = aa;
        idNumber = ii;
    }

    void print() {
        cout << "\nStudent name: " << name
        << "\n\tAge: " << age
        << "\n\tID: " << idNumber << endl;
    }
private:
    string name;
    int age;
    int idNumber;
};

int main()
{
    Students student01("KJ", 20, 1000);
    Students student02("Tom", 21, 1001);
    Students student03("Julie", 19, 1002);
    student01.print();
    student02.print();
    student03.print();
    cout << "\nClass object variable in array style:" << endl;
    Students s[3] = { Students("Pokemon",25,1003), Students("Digimon",20,1004), Students("Barbie",26,1005) };
    for (int i = 0; i < 3; i++)
        s[i].print();
    return 0;
}
```

```
Student name: KJ
        Age: 20
        ID: 1000

Student name: Tom
        Age: 21
        ID: 1001

Student name: Julie
        Age: 19
        ID: 1002

Class object variable in array style:

Student name: Pokemon
        Age: 25
        ID: 1003

Student name: Digimon
        Age: 20
        ID: 1004

Student name: Barbie
        Age: 26
        ID: 1005
```

80 points

- Your program must include
  1. Structure (or class) design
     - Must design structures (or classes) such as person, player, referee, game class etc.
  2. Method/function blocks
     - Design your own methods (e.g., scoring, penalty kick etc.)
  3. User inputs, Loops & Conditional statements
     - Being used to design your game logics
  4. Arrays (including an array of objects) and Pointer variable (*)
     - Must demonstrate how to use both primitive array data and an array of objects
  5. Game logics (2 Semi-finals & Final game)
     - 3 games.
  6. Randomization for base scoring/performance
     - Review previous assignment (also next slide)
  7. Adding Penalty kick mechanism
     - Computer vs. User choice. Make a game more fun and dynamic through a user interaction.
  8. File generation (summary result output to both screen and txt format)
     - Generate a txt file to show the summary (winner, total score, MVP etc. Be creative)
     - Display ranking in sorted order.
     - Review lecture 5

Project requirement:

1. Your project is due by 8/21 11:59 pm. No late submission will be accepted.
2. This is an individual assignment, no plagiarism. If more than 20% of your code looks similar/same from another person's work, your work will be considered as 'copying, violating plagiarism', and both people (who provides codes and you) will receive '0' on this project.
3. You must submit one zip file including all your C++ files (test it before you submit!) and one MS Word document (minimum 3 pages, name it as project_youLastnameFirstnameInitial.docx)
   - In your MS Word document,
     - Provides overall summary of your design approach
     - summarize the highlights of your work including your unique design features
     - indicates things/reason/constraints which you weren't able to accomplish (even if some of your program lacks required components, you may get some partial marks by explaining why/what went wrong)
     - UML diagram
4. Please review all the specification describe in next pages.
5. Some bonus mark will be added on your extra work.

Team and Player data generation

Generate 2 semi-final games and one final game,

a) For the final game, two winning teams from the semi-final game should battle for the championship!

b) Using array(s) (1D or 2D), create minimum 4 teams, and define minimum 3 variables/factors characterizing team performance (i.e., Nationality reputation, Speed, Power, Injury, Teamwork and Star Player)

- Game play interface: When two teams for each game are selected, you program must display both name of each country(or team) and name of all 4 players with each player's (minimum) 3 skill levels.
  - Also, when the game ends, the game statistics/results should both be displayed on the monitor and be generated as 'wcResult.txt' file showing the summary (3 games: score of each team from 2 semi-final games, and the final game) This topic (file read/write) will be covered next week.
  - Your program should display the record of all 4 teams, and output the team ranks in order based on each member variable, not total. Also, calculate the total sum for each team, and show the winning team. Also, try an extra factor (quality of star player)

Game mechanics/play

1. Functions relate to game play

a) Scoring (goal!) can be achieved by comparing performance of team players from two different teams:

b) Design functions relate to attack and defense functions.

- Not all 4 players should get involved in attack or defense. Pick a random number of players who will be involved in attack and defense. Then compare their total skill sets on top of overall team reputation/performance. You must design your own formula to figure out combined skill sets of multiple players. Also, consider how you can incorporate overall team reputation/performance into player's performance.
    - Example: Aassign a random value (1 as poor -5 as best) to 3 variables (speed, power, and/or teamwork) and 2 user entered values (Nationality and Star Player) relate to performance and apply these values to performance formula you designed (e.g. Performance = 2 x Nationality + Speed x Power + Teamwork + Star Player - injury) .
    - Could However, the magnitude of selected weight variable is not known to the user (Surprise! Apply random selected weight values for each team). If less than 1 (multiply this weight value to a selected variable; recalculation!), under performance than previous initial sum. If more than 1 (multiplied by weight >1), this special weight value introduced on one variable would perform better than the normal weight (1.0).

a. Design functions relate to penalty function as well as a function representing a unique skill to goalkeeper (e.g., catchingBall).

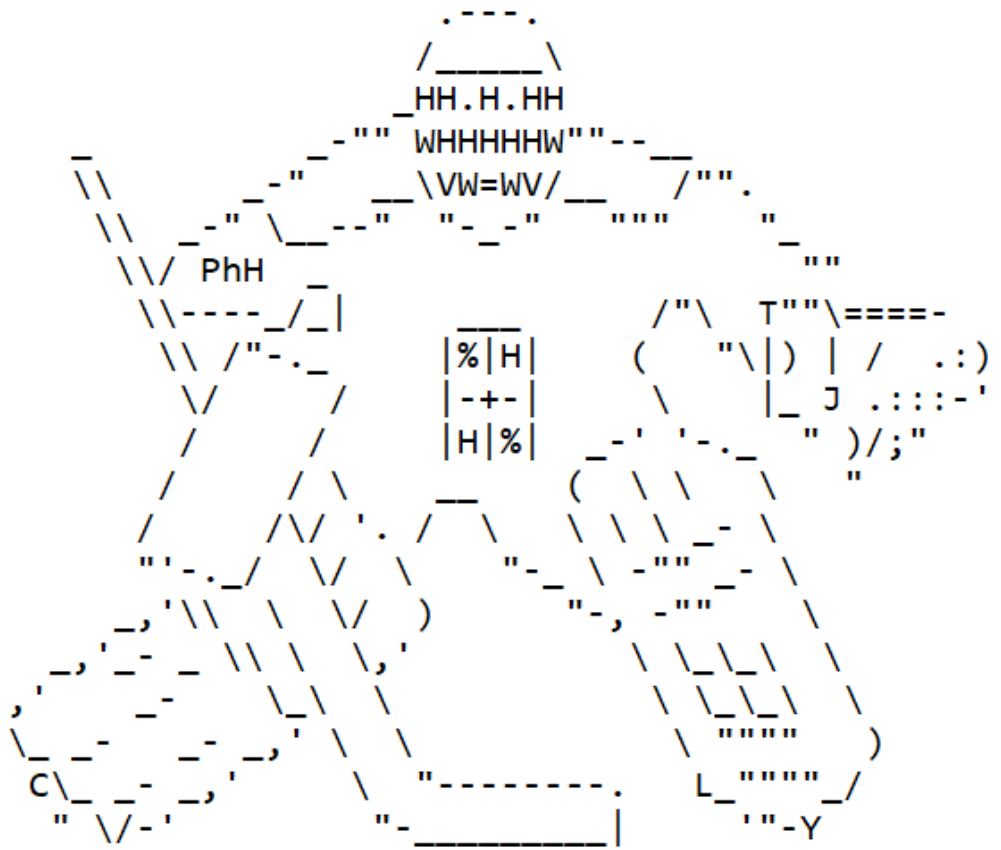
b. Design functions relate to foul, and receiving Yellow or Red card (Link explaining foul and yellow/red card from referee : http://www.understandingsoccer.com/rule-12-fouls-and-misconduct.html). Red card removes a player from the game.

1. Using array(s) (1D or 2D), create minimum 4 teams, and define 5 variables/factors(3 random values and 2 value entered by a user) characterizing team performance (i.e., Nationality reputation, Speed, Power, Injury, Teamwork and Star Player)
2. To compare initial performance of each team, assign a random value (1 as poor -5 as best) to 3 variables (speed, power, and/or teamwork) and 2 user entered values (Nationality and Star Player) relate to performance and apply these values to performance formula you designed (e.g. Performance = 2 x Nationality + Speed x Power + Teamwork + Star Player - injury) .
3. Team competition is based on the total sum of all member variables for each team. To make the game more engaging, apply one weight value directly to manipulate/change one specific member variable entered by the user(either speed, power or teamwork) affecting initial sum.
4. However, the magnitude of selected weight variable is not known to the user (Surprise! Apply random selected weight values for each team). If less than 1 (multiply this weight value to a selected variable; recalculation!), under performance than previous initial sum. If more than 1 (multiplied by weight >1), this special weight value introduced on one variable would perform better than the normal weight (1.0).
5. Your program should display the record of all 4 teams, and output the team ranks in order based on each member variable, not total. Also, calculate the total sum for each team, and show the winning team. Also, try an extra factor (quality of star player)

Tips for Bonus mark! Add more factors affecting game play and user interaction.

- Research what typically happens at the championship? Who can be a MVP? Audience factor?
- Any extra factor affecting game condition? Different locations? Weather? Home advantage?
- Feel free to add more detail/refinement on certain functions (e.g., different attack types?)

--------WELLCOME TO WORLD CUP HOCKEY 2016--------



```
Set Team Canada's skill level from (1-5) 1
Set Team Russia's skill level from (1-5) 2
Set Team USA's skill level from (1-5) 3
Set Team Poland's skill level from (1-5) 4
Wanna Play shootout round or Game
Enter 1 for Shootout    2 for Game
2
Enter Player's skill level in Scale of (1-5) for Sidney Crosby: 3
Enter player Sidney Crosby aggression in scale(1-5): 2
Enter Player's skill level in Scale of (1-5) for Roberto Luongo: 3
Enter player Roberto Luongo aggression in scale(1-5): 2
Enter Player's skill level in Scale of (1-5) for John Tavares: 5
Enter player John Tavares aggression in scale(1-5): 4
Enter Player's skill level in Scale of (1-5) for Wayne Gretzky: 2
Enter player Wayne Gretzky aggression in scale(1-5): 1
Enter Player's skill level in Scale of (1-5) for Alexander Ovechkin: 3
Enter player Alexander Ovechkin aggression in scale(1-5): 2
Enter Player's skill level in Scale of (1-5) for Evgeni Malkin: 1
Enter player Evgeni Malkin aggression in scale(1-5): 2
Enter Player's skill level in Scale of (1-5) for Pavel Datsyuk: 3
Enter player Pavel Datsyuk aggression in scale(1-5): 5
Enter Player's skill level in Scale of (1-5) for Pavel Bure: 2
Enter player Pavel Bure aggression in scale(1-5): 4
Enter Player's skill level in Scale of (1-5) for Ryan Kesler: 3
Enter player Ryan Kesler aggression in scale(1-5): 2
Enter Player's skill level in Scale of (1-5) for Patrick Kane: 1
Enter player Patrick Kane aggression in scale(1-5): 3
Enter Player's skill level in Scale of (1-5) for Joe Pavelski: 1
Enter player Joe Pavelski aggression in scale(1-5): 3
Enter Player's skill level in Scale of (1-5) for Phil Kessel: 2
Enter player Phil Kessel aggression in scale(1-5): 3
Enter Player's skill level in Scale of (1-5) for Henryk Gruth: 2
Enter player Henryk Gruth aggression in scale(1-5): 1
Enter Player's skill level in Scale of (1-5) for Krzysztof Oliwa: 2
Enter player Krzysztof Oliwa aggression in scale(1-5): 3
Enter Player's skill level in Scale of (1-5) for Peter Sidokiewicz: 2
Enter player Peter Sidokiewicz aggression in scale(1-5): 1
Enter Player's skill level in Scale of (1-5) for Mariusz Czerkawski: 3
Enter player Mariusz Czerkawski aggression in scale(1-5): 2
```

Well Come to World Cup Hockey 2016
The 4 teams that are in Semi Finals are:

1. Canada
2. Russia
3. USA
4. Poland

---------- Each Team with Member Details ----------

-----Team Canada -----

S.No.	Name	endurance	athleticism	Teamwork	Performance
1.	Sidney Crosby	1	1	1	10
2.	Roberto Luongo	2	5	4	22
3.	John Tavares	5	5	4	43
4.	Wayne Gretzky	4	2	2	15

-----Team Russia -----

S.No.	Name	endurance	athleticism	Teamwork	Performance
1.	Alexander Ovechkin	3	3	1	18
2.	Evgeni Malkin	1	5	4	13
3.	Pavel Datsyuk	1	4	1	16
4.	Pavel Bure	5	5	3	36

-----Team USA -----

S.No.	Name	endurance	athleticism	Teamwork	Performance
1.	Ryan Kesler	1	1	5	14
2.	Patrick Kane	5	3	4	24
3.	Joe Pavelski	5	1	2	12
4.	Phil Kessel	3	4	3	22

-----Team Poland -----

S.No.	Name	endurance	athleticism	Teamwork	Performance
1.	Henryk Gruth	1	5	1	11
2.	Krzysztof Oliwa	3	5	3	25
3.	Peter Sidokiewicz	5	1	1	11
4.	Mariusz Czerkawski	2	2	2	14

----- First Semi Final -----

Match Between Team Canada and Team Russia

Teams are ready Enter s to Start: s

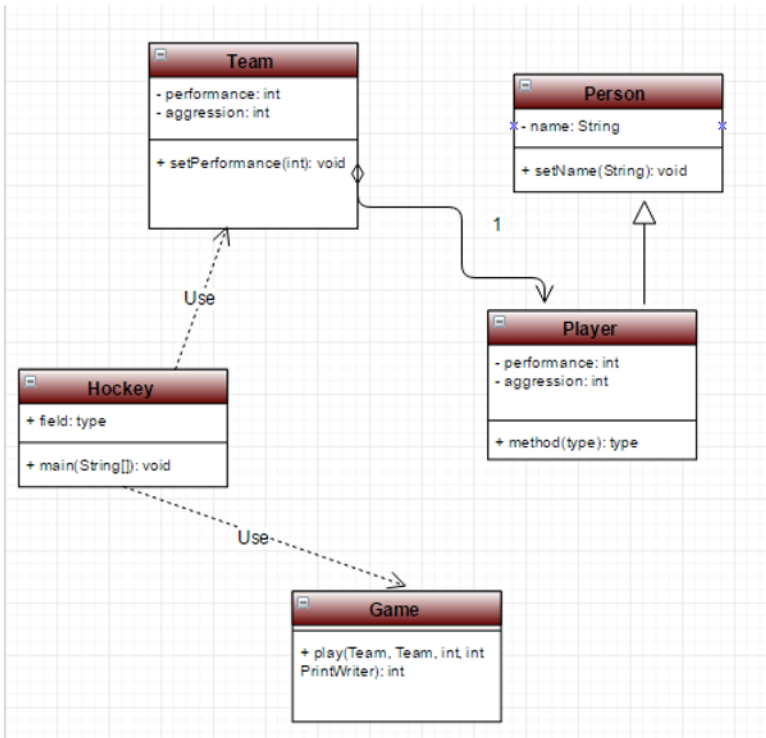
Team Canada is attacking Russia
Attacking
Team Canada Score a Goal
Attacking
Team Canada Score a Goal
Attacking
Attacking
Attacking
Team Canada Score a Goal
Attacking
Attacking
Team Canada Score a Goal
Foul from Canada Player
Attacking
Team Russia is attacking Canada
Attacking
Attacking
Attacking
Attacking
Attacking
Attacking
Attacking
Team Russia Score a Goal
Foul from Russia Player
Attacking
Canada = 4 Goals
Russia = 1 Goals

Team Canada won the Game

```
----- Second Semi Final -----

Match Between Team USA and Team Poland

Teams are ready Enter s to Start: s


 Team USA is attacking Poland
                  Attacking
                  Attacking
Team USA Score a Goal
                  Attacking
                  Attacking
Team USA Score a Goal
                  Attacking
                  Attacking
Team USA Score a Goal
                  Attacking
                  Attacking
                  Attacking
                  Attacking
                  Attacking
Team USA Score a Goal
                  Attacking
                  Attacking


 Team Poland is attacking USA
                  Attacking
                  Attacking
Team Poland Score a Goal
                  Attacking
                  Attacking
                  Attacking
                  Attacking
Team Poland Score a Goal
                  Attacking
                  Attacking
                  Attacking
                  Attacking
 Foul from Poland Player
                  Attacking
Team Poland Score a Goal
                  Attacking
                  Attacking
Team Poland Score a Goal
USA = 4 Goals
Poland = 4 Goals

Team Poland won the Game
```

```
----- Final Match -----

Match Between Team Canada and Team Poland

Teams are ready Enter s to Start: s


 Team Canada is attacking Poland
                    Attacking
                    Attacking
Team Canada Score a Goal
                    Attacking
                    Attacking
                    Attacking
                    Attacking
                    Attacking
                    Attacking
Team Canada Score a Goal


 Team Poland is attacking Canada
                    Attacking
                    Attacking
                    Attacking
                    Attacking
                    Attacking
Team Poland Score a Goal
                    Attacking
                    Attacking


Team Poland Score a Goal
                   Attacking
Canada = 2 Goals
Poland = 2 Goals

Team Poland won the Game


----- Final Result -----
The Euro Cup is ended!! Team Poland is the winner
```

```
 ---- Details of Teams Performance and Points ----

S.No. Team Name        Total Goals     Total Match Wins     Total Points
1.    Canada            6                 1                    40
2.    Russia            1                 0                    27
3.    USA               4                 0                    27
4.    Poland            6                 2                    63


 ---- Bye Bye ----
```



Well Come to World Cup Hockey
 The 4 teams that are in Semi Finals are:
The 4 teams that are in Semi Finals are:

1. Canada
2. Russia
3. USA
4. Poland

----- First Semi Final -----

Match Between Team Canada and Team Russia
Canada = 4 Goals
Russia = 1 Goals

Team Canada won the Game

----- Second Semi Final -----

Match Between Team USA and Team Poland
USA = 4 Goals
Poland = 4 Goals
Team Poland won the Game

----- Final Match -----

Match Between Team Canada and Team Poland
Canada = 2 Goals
Poland = 2 Goals
Team Poland won the Game

----- Final Result -----

The Euro Cup is ended!! Team Poland is the winner

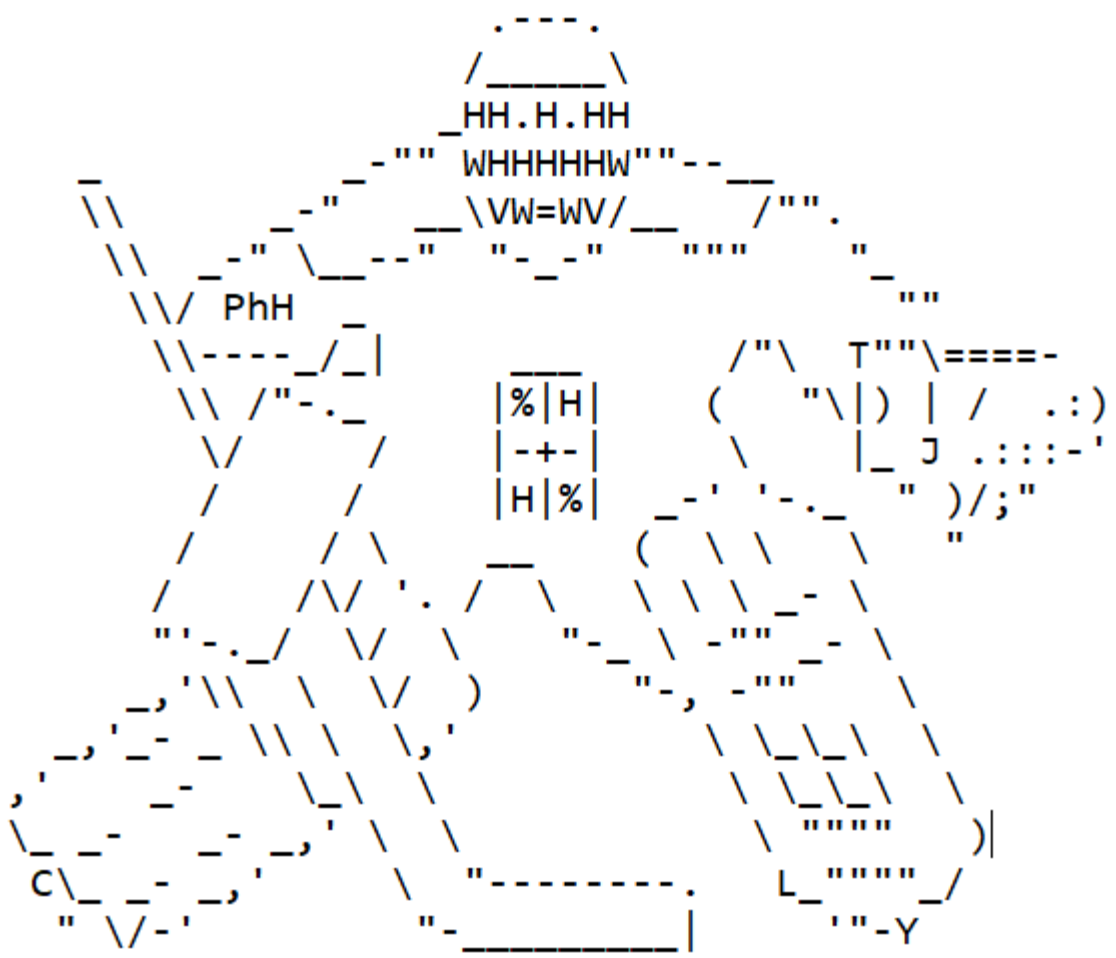
 ---- Details of Teams Performance and Points ----

S.No.	Team Name	Total Goals	Total Match Wins	Total Points
1.	Canada	6	1	40
2.	Russia	1	0	27
3.	USA	4	0	27
4.	Poland	6	2	63

 ---- Bye Bye ----

International Hockey Results.txt

--------WELLCOME TO WORLD CUP HOCKEY 2016--------



Set Team Canada's skill level from (1-5) 2
Set Team Russia's skill level from (1-5) 1
Set Team USA's skill level from (1-5) 3
Set Team Poland's skill level from (1-5) 4
Wanna Play shootout round or Game
Enter 1 for Shootout   2 for Game
1

Canada is attacking

Where do you want to shoot Select a placement:

Press 1 for left, 2 for right, 3 for middle

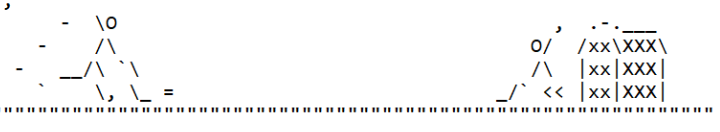
1
Nice Shot!! Its a Goal
Where do you want to shoot Select a placement:

Press 1 for left, 2 for right, 3 for middle

2
Nice Shot!! Its a Goal

Where do you want to shoot Select a placement:

Press 1 for left, 2 for right, 3 for middle



3
You Missed!
Your total score is: 2
Now it is opposing team's turn
Your time to defend. Be ready!

Where do you want to Defend!Select a placement:

Press 1 for left, 2 for right, 3 for middle

```
Where do you want to Defend!Select a placement:

Press 1 for left, 2 for right, 3 for middle
 ,
      -  \O                                                  ,  .-.___
    -     /\                                               O/  /xx\XXX\
  -    __/\ `\                                             /\  |xx|XXX|
    `     \, \_ =                                       _/` << |xx|XXX|
""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""

1
Opposing team scored! Try harder

Where do you want to Defend!Select a placement:

Press 1 for left, 2 for right, 3 for middle
 ,
      -  \O                                                  ,  .-.___
    -     /\                                               O/  /xx\XXX\
  -    __/\ `\                                             /\  |xx|XXX|
    `     \, \_ =                                       _/` << |xx|XXX|
""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""

2
Opposing team scored! Try harder

Your score: 2. Opposing team's score: 3
You lost! Better luck next time!
Wanna Play again!
 Enter 1 for Play again 2 for exit:
```

Penalty kick mechanism

```
-----------------------------------------------
WEST HAM UNITED is attacking.
-----------------------------------------------
|  1                  2                  3    |
|  4                  5                  6    |
|  7                  8                  9    |
-----------------------------------------------
where do you want to shoot? Enter number:4
NICE! GOAL!!
```

```
MANCHESTER CITY is defending
-----------------------------------------------
|  1                  2                  3    |
|  4                  5                  6    |
|  7                  8                  9    |
-----------------------------------------------
Where do you think the computer is going to shoot?  Enter number:2
Too bad! You could't defend!
```

```
----------------------Displaying Premier League Teams---


1.ARSENAL
2.CHELSEA
3.MANCHESTER UNITED
4.WEST HAM UNITED
5.MANCHESTER CITY
6.LIVERPOOL
```

```
-------------------------------------------------------------------------
ARSENAL
-------------------------------------------------------------------------
Andrea Pirlo                Q       14      86      14      67
Philipp Lahm               MX       77      46      70      68
Mats Hummels               BD       34      67      94      9
Oscar                  NT   5       2       100     17
Giorgio Chiellini          JO       3       41      33      71
Mathieu Valbuena           MY       4       15      14      100
Karim Benzema              J        6       13      14      15
Franck Ribery              TF       92      22      60      50
Gianluigi Buffon           VY       3       88      50      29
Marcelo                ZK   68      78      66      46
Victor Wanyama             SO       57      27      74      66
```

```
Match Between team 1 and team 2:



Team CHELSEA is attacking the Team MANCHESTER UNITED.
                                    Attacking
                                    Attacking
                                    Attacking

Team ARSENAL scored.
 Total attack: 606 Total defence: 300
Team MANCHESTER UNITED is attacking the Team CHELSEA.
                                    Attacking
                                    Attacking
                                    Attacking
                                    Attacking
                                    Attacking

Team CHELSEA scored.
```

```
Team MANCHESTER UNITED won the game
```

```
Semi Final number 2:
-----------------------------------------------------------
Match Between team WEST HAM UNITED and team MANCHESTER CITY :



Team WEST HAM UNITED is attacking the Team MANCHESTER CITY.

                                        Attacking
                                        Attacking
                                        Attacking
                                        Attacking
                                        Attacking
                                        Attacking
                                        Attacking

NICE defend!!   None of the teams gets a score.
Team MANCHESTER CITY is attacking the Team WEST HAM UNITED.
                                        Attacking
                                        Attacking
                                        Attacking
                                        Attacking
                                        Attacking
                                        Attacking
```

```
------------------------FINAL-------------------------------
Match Between team MANCHESTER UNITED and team MANCHESTER CITY :



Team MANCHESTER UNITED is attacking the Team MANCHESTER CITY.
                                        Attacking
                                        Attacking
                                        Attacking

Team MANCHESTER UNITED scored.
 Total attack: 631 Total defence: 622
Team MANCHESTER CITY is attacking the Team MANCHESTER UNITED.
                                        Attacking
                                        Attacking
                                        Attacking
                                        Attacking
                                        Attacking

Team MANCHESTER CITY scored.
```

```
The final Result-----------------
The Premier League is END!! Team: MANCHESTER CITY is the winner!!!
```

Problem Solving

Problem solving

The act of finding a solution to a perplexing, distressing, vexing, or unsettled question

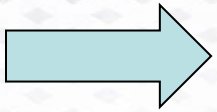
*How do **you** define problem solving?*

Problem Solving

How to Solve It: A New Aspect of Mathematical Method by George Polya

"How to solve it list" written within the context of mathematical problems

But list is quite general



We can use it to solve computer related problems!

Problem Solving

How do you solve problems?

Understand the problem

Devise a plan

Carry out the plan

Look back

Strategies

Ask questions!

- *What do I know about the problem?*
- *What is the information that I have to process in order the find the solution?*
- *What does the solution look like?*
- *What sort of special cases exist?*
- *How will I recognize that I have found the solution?*

Strategies

Ask questions! Never reinvent the wheel!

Similar problems come up again and again in different guises

A good programmer recognizes a task or subtask that has been solved before and plugs in the solution

Can you think of two similar problems?

Strategies

Divide and Conquer!

Break up a large problem into smaller units and solve each smaller problem

- Applies the concept of abstraction
- The divide-and-conquer approach can be applied over and over again until each subtask is manageable

Computer Problem-Solving

- Analysis and Specification Phase
  - Analyze
  - Specification
- Algorithm Development Phase
  - Develop algorithm
  - Test algorithm
- Implementation Phase
  - Code algorithm
  - Test algorithm
- Maintenance Phase
  - Use
  - Maintain

Can you name a recurring theme?

Phase Interactions

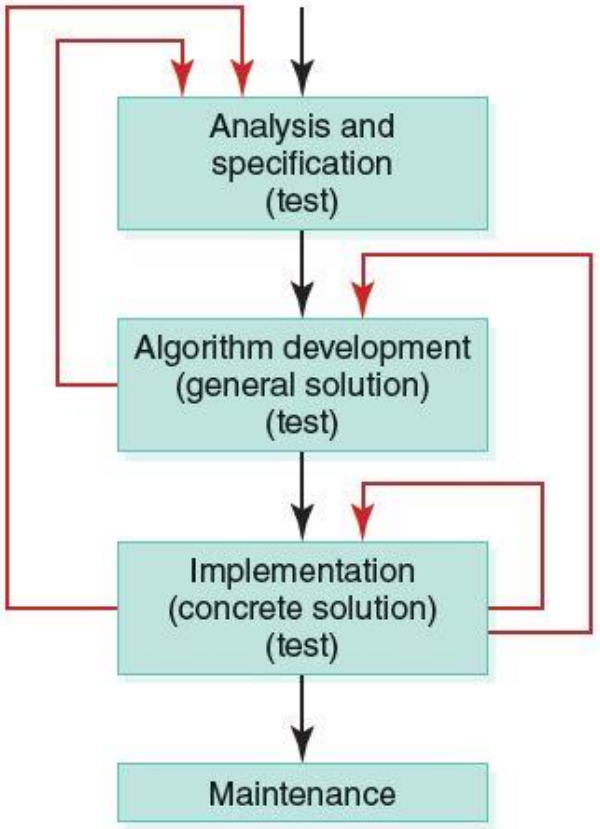


FIGURE 7.3 The interactions among the four problem-solving phases

Should we add another arrow?

(What happens if the problem is revised?)

Algorithms

Algorithm

A set of unambiguous instructions for solving a problem or subproblem in a finite amount of time using a finite amount of *data*

Abstract Step

An algorithmic step containing unspecified details

Concrete Step

An algorithm step in which all details are specified

Review
C++ class (Chapter 2)

C++ Classes

C++ Supports the use of classes to define new data types.

- Definition of a new class type requires a
  - Class Declaration
  - Class Implementation

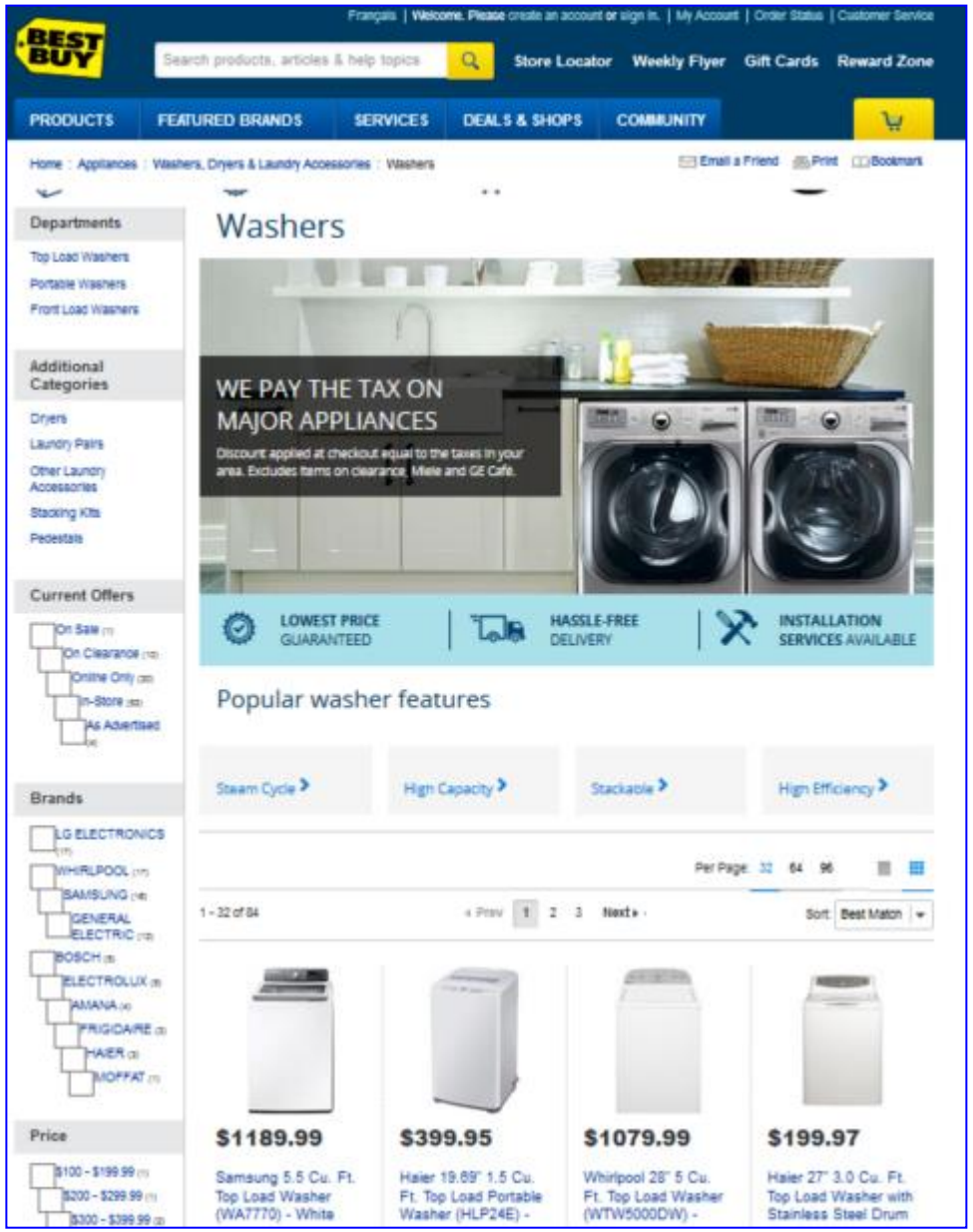
Object-Oriented Programming

- **Object-oriented programming: focused on creating objects**
- **Object: entity that contains data and procedures**
  - Data is known as data attributes and procedures are known as methods
    - Methods perform operations on the data attributes
- **Encapsulation: combining data and code into a single object**

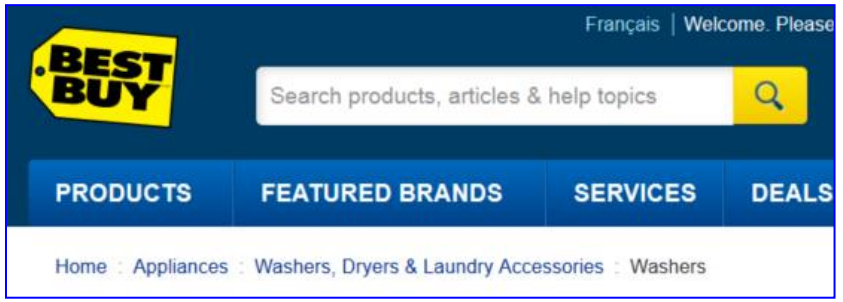
Unified Modeling Language (UML)

- The UML is a visual modeling language that enables system builders to create blueprints that capture their visions in a standard, easy-to-understand way, and provides a mechanism to effectively share and communicate these visions with others.

The key is to organize the design process in a way that analysts, clients, programmers, and others involved in system development can understand and agree on.
The UML provides the organization.



Washing Machine



http://www.bestbuy.ca/en-CA/category/washers/33931p.aspx

Overview | Details & Specs

Overview

Who would have thought that doing laundry could be so easy? With the LG WM9000HVA mega-capacity 6.0 cu. ft. washer you can easily deal with more laundry in a shorter amount of time. LG's TurboWash shortens laundry day even more, saving you up to 30 minutes per cycle. It has 14 programmed cycles and more that you can download with the free app.

More Information

No longer a chore

- Whether you need to tackle a mountain of towels or heaps of jeans, the 6.0 cu. ft. capacity can do more in one load so you spend less time doing laundry.
- With LG TurboWash you get exceptional cleaning performance while saving you time, up to 30 minutes per load. Dual nozzles spray clothes with a concentrated detergent solution that cleans your laundry faster, and high-pressure nozzles rinse clothes faster and more efficiently during the high-speed rinse.

Hot and cold

- LG Steam technology delivers enhanced cleaning by more effectively penetrating fabric to remove dirt and soil. With the Allergiene cycle common household allergens don't stand a chance. Certified by the Asthma Society of Canada (ASC), it removes over 95% of common allergens such as dust mites and pet dander.
- With ColdWash technology you can confidently use the cold cycle without sacrificing cleaning power. It uses enhanced washing motions with cold water to give you warm-water cleaning performance every time.
- It boasts 14 programmed wash cycles, including Cotton/Normal, Bulky/Bedding, Allergiene, Heavy Duty, BrightWhites, and more; 12 wash options such as Prewash, ColdWash, Steam, TurboWash, and more; and 5 water temperature settings, so you always get the right setting for whatever your laundry demands.

Eco-friendly

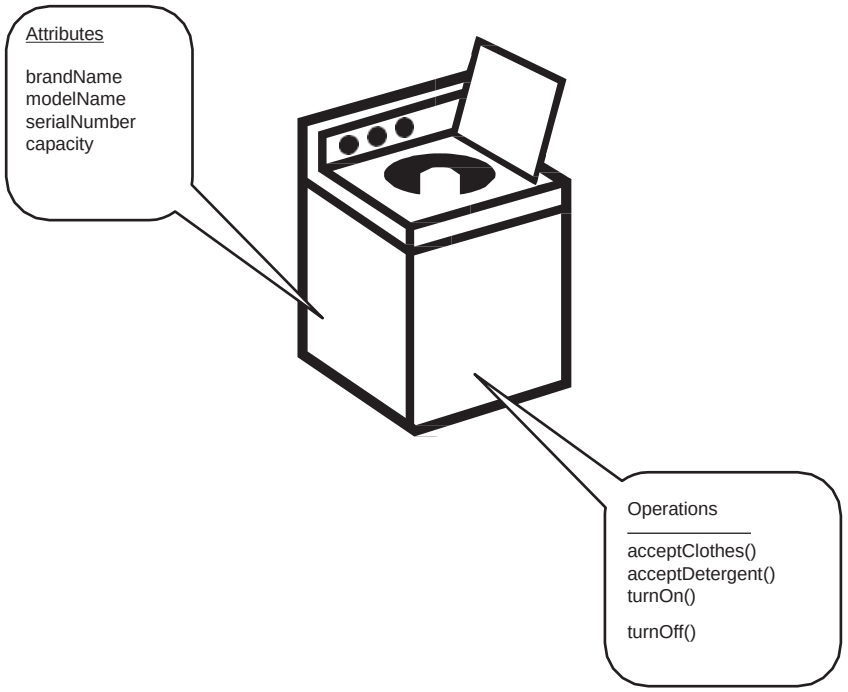
- ENERGY STAR qualified means this washer works to conserve natural resources and help save you money on utility bills. Water levels adjust automatically to load size so it doesn't waste water. Plus, ColdWash and TurboWash cut down on energy so you and the planet share in the savings.
- With integrated Smart Grid technology, this LG washer is able to detect when power use in your area is at its lowest, so your appliance can operate at lower energy rates.

Overview | Details & Specs

Capacity (Cu. Ft.)	6.0 cu ft
Control Configuration	Touch
Colour Family	Silver
Colour	Silver
Load Access	Front Load
Control Panel Material	Plastic
Power Source Requirement	120 V; 10 amps
High Efficiency	Yes
EnerGuide Rating (KWH/Year)	150 kWh/yr
Energy Star Qualified	Yes
Stackable	No
Wash Features	
Cycle Descriptions	Cotton/Normal; Perm. Press; Delicates; Speed Wash; Tub Clean; Sanitary; Allergiene; Heavy Duty; Bright Whites; Towels; Rinse+Spin; Download; Jumbo Wash; Bedding
Number of Wash Cycles	14
Water Temperature Options	5
Wash Basket Material and Interior Finish	Never Rust Stainless Steel Drum
Noise Reduction	Yes
Sound Insulation Package	LoDecibel Quiet Operation
Wash Option	Prewash; Delay Wash; Extra Rinse; Child Lock; ColdWash; FreshCare; Steam; TurboWash; Custom Program; Drum Light; Smart Grid; Remote Start
Spin Speed (RPM)	1300 rpm
Washer Drive System	Direct

http://www.bestbuy.ca/en-CA/product/lg-electronics-lg-29-6-0-cu-ft-front-load-washer-wm9000hva-silver-wm9000hva/10397331.aspx?path=d35ca49232f245250683b52917cad954en02

WashingMachine class is a template for creating new instances of washing machines.



Class Diagram

WashingMachine
brandName modelName serialNumber capacity
acceptClothes() acceptDetergent() turnOn() turnOff()

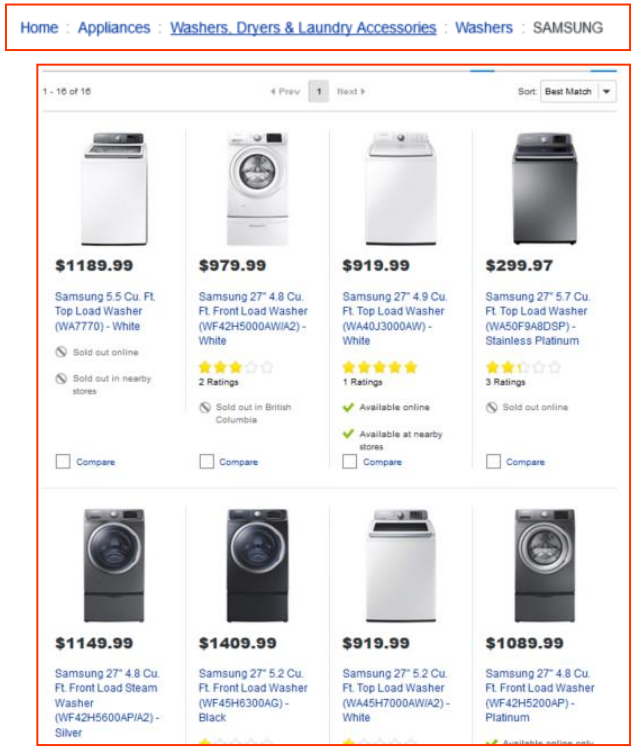
Object Diagram

myWasher:WashingMachine	:WashingMachine

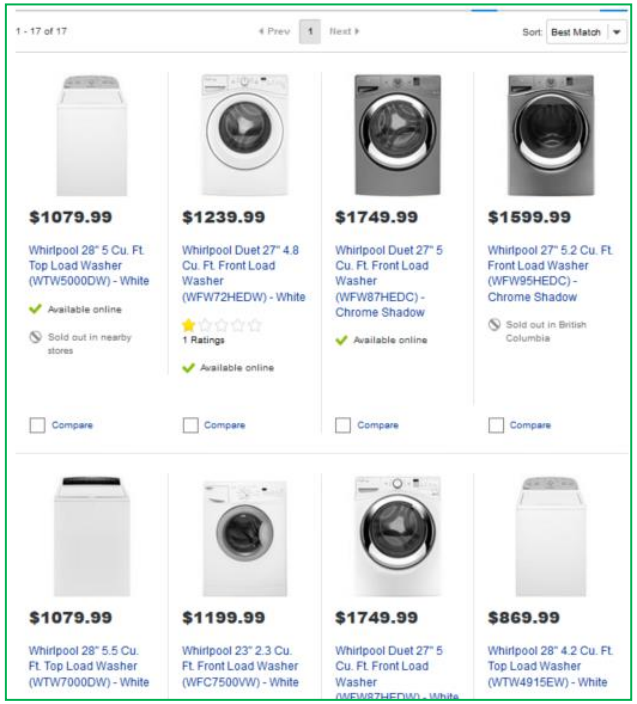
Object Diagram

myWasher:WashingMachine

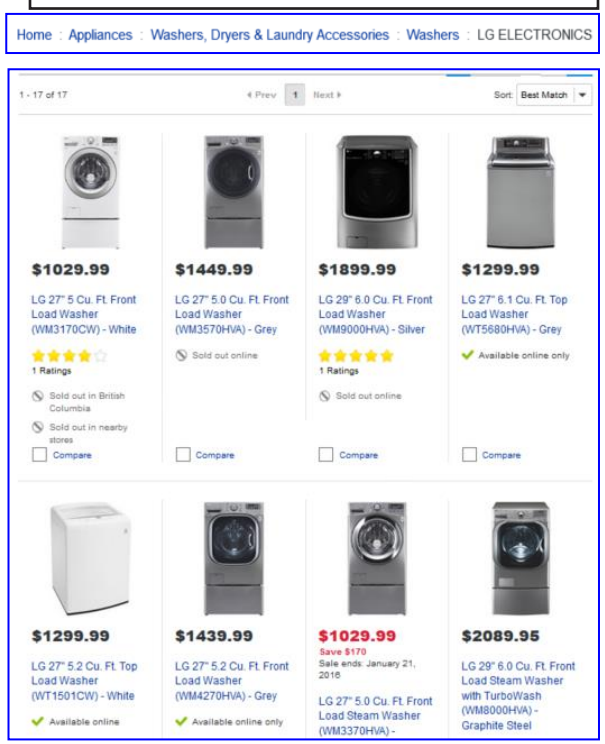
Samsung:WashingMachine

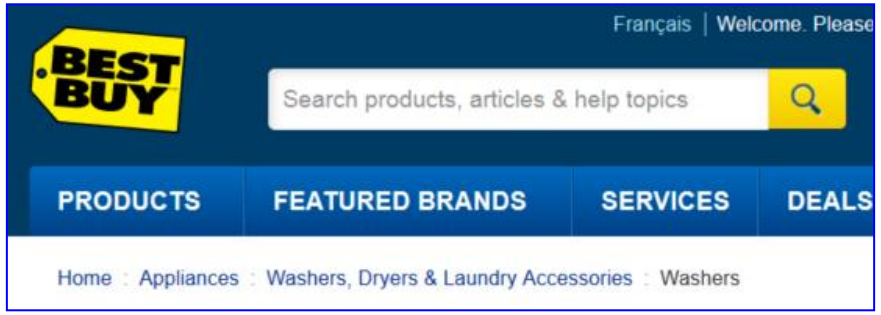


Whirlpool:WashingMachine

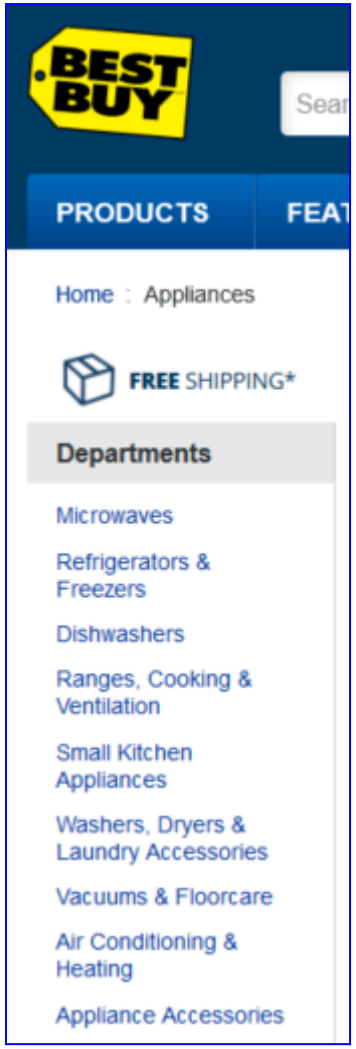


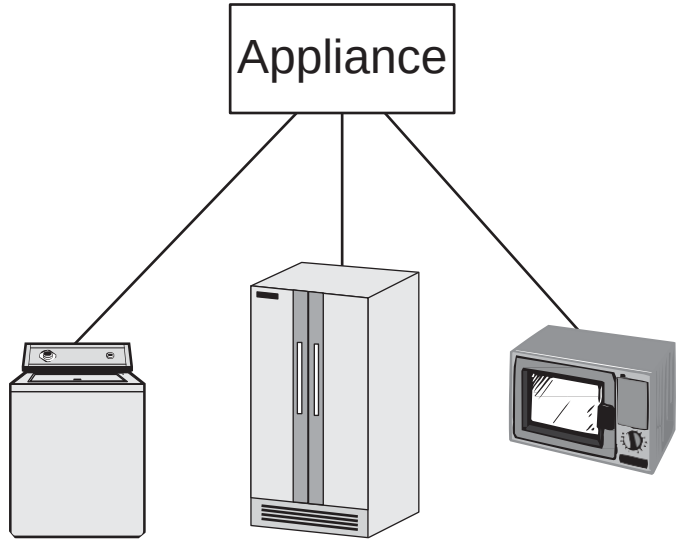
LG:WashingMachine



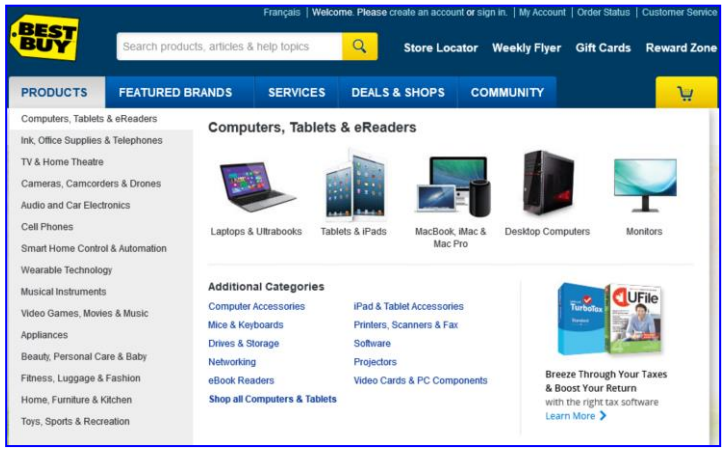


Superclass & subclass

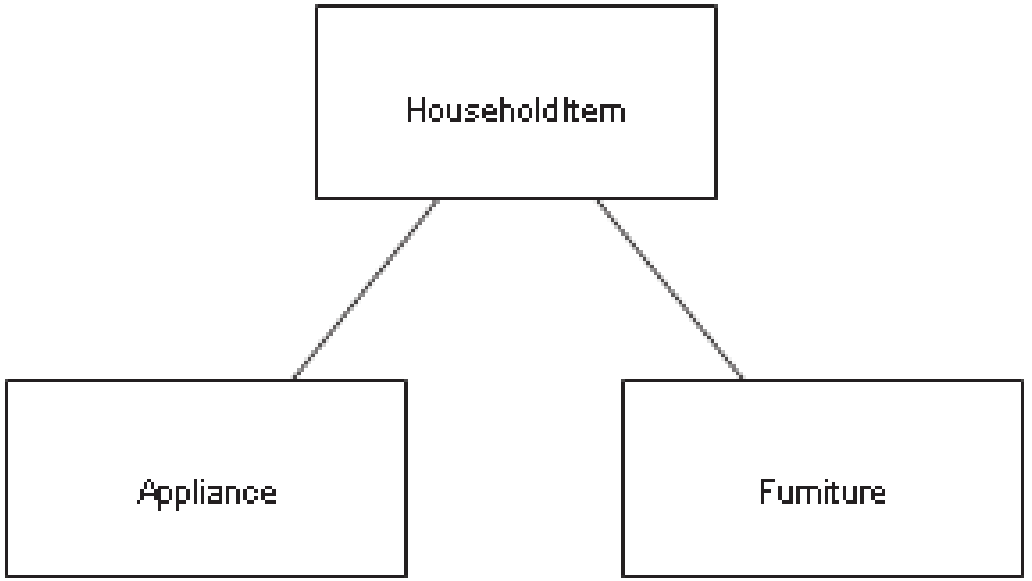


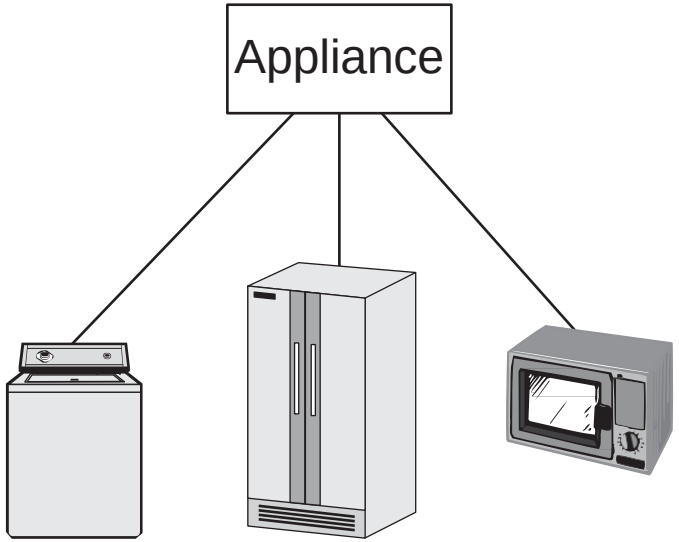


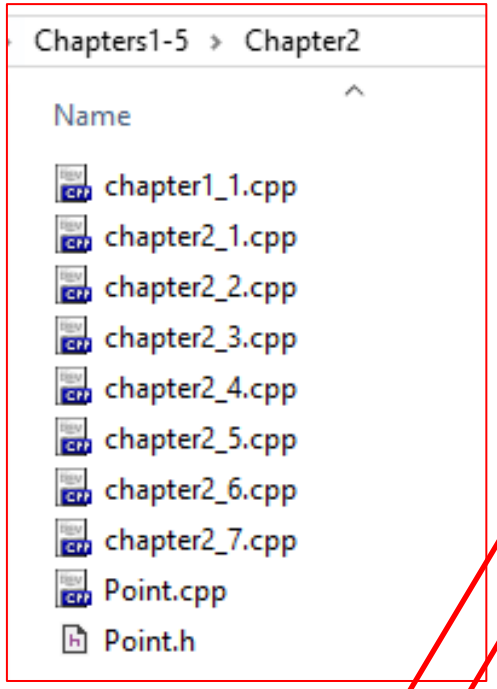
Appliances inherit the attributes and operations of the Appliance class. Each one is a subclass of the Appliance class. The Appliance class is a superclass of each subclass.



Superclasses can also be subclasses and inherit from other superclasses.

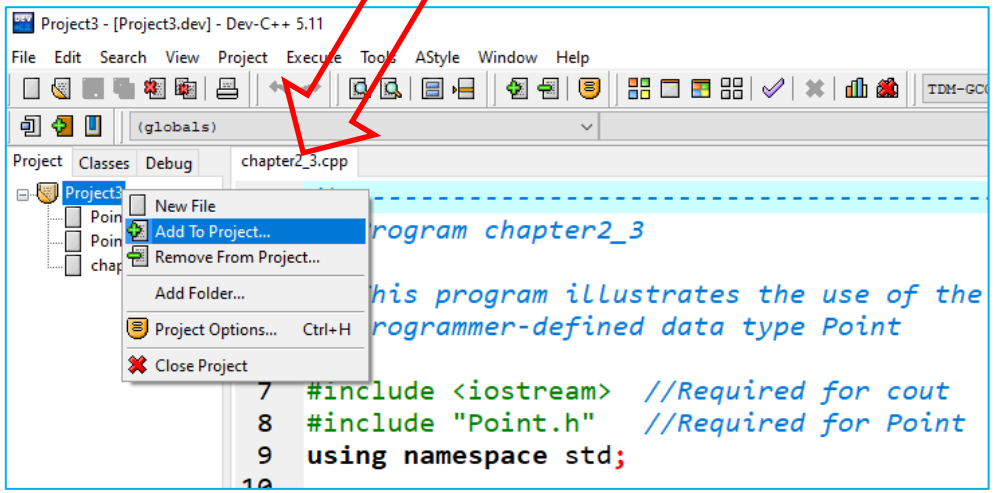






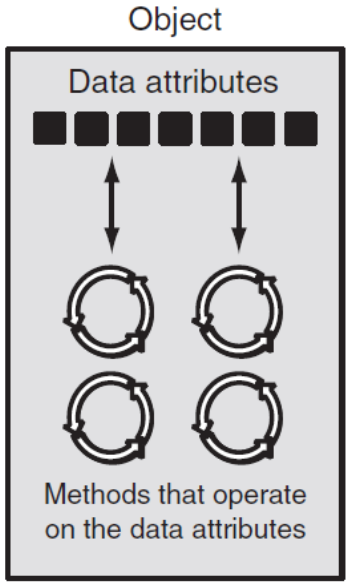
To run chapter 2.3 example from Textbook source code

1. Create a new project by choosing File > New > Project. Choose Empty project (or Console window).
2. Remove any existing file from the project
3. Press/hold RMB (right mouse button click), and select Add to Project option.
4. Choose three files (chapter2_3.cpp, Point.cpp and Point.h)



Object-Oriented Programming (cont'd.)

Figure 10-1 An object contains data attributes and methods

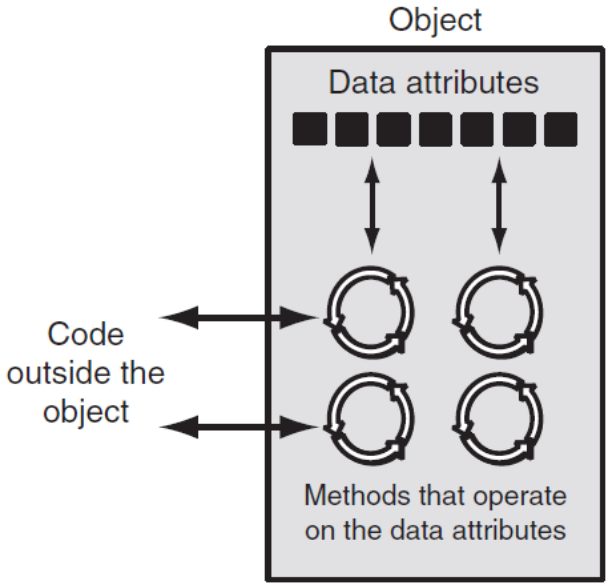


Object-Oriented Programming (cont'd.)

- **Data hiding: object's data attributes are hidden from code outside the object**
  - Access restricted to the object's methods
    - Protects from accidental corruption
    - Outside code does not need to know internal structure of the object
- **Object reusability: the same object can be used in different programs**
  - Example: 3D image object can be used for architecture and game programming

Object-Oriented Programming (cont'd.)

Figure 10-2 Code outside the object interacts with the object's methods

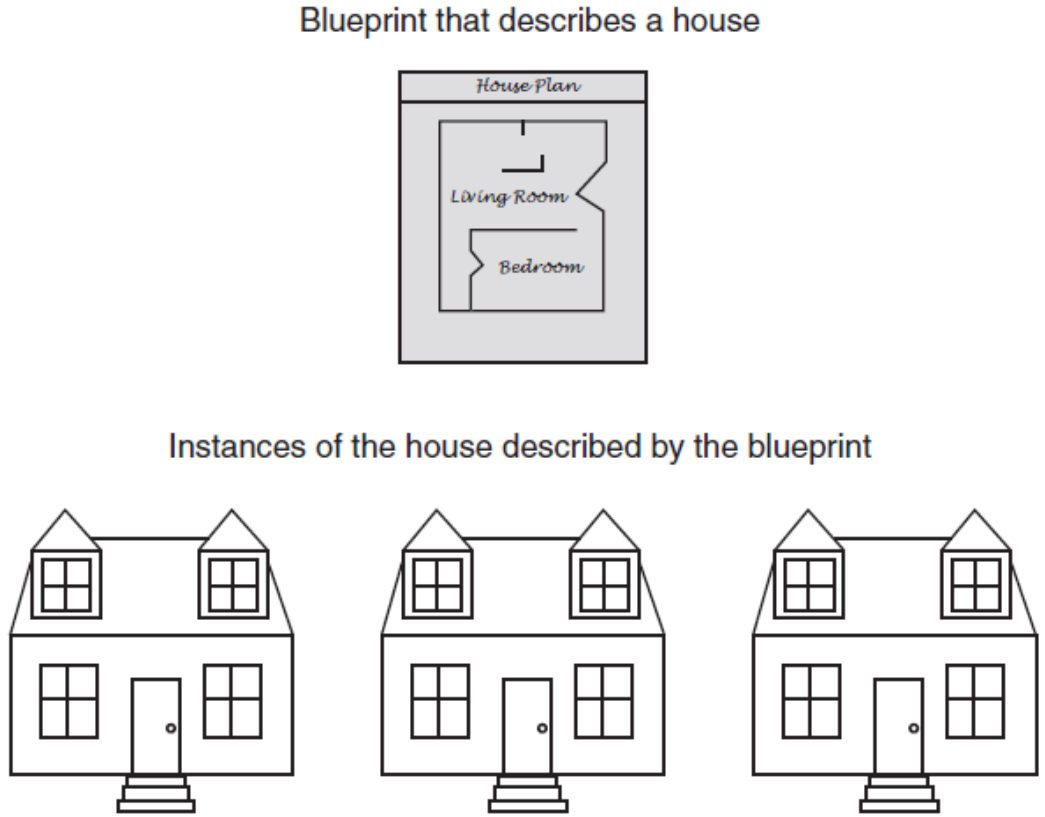


Classes

- **Class: code that specifies the data attributes and methods of a particular type of object**
  - Similar to a blueprint of a house or a cookie cutter
- **Instance: an object created from a class**
  - Similar to a specific house built according to the blueprint or a specific cookie
  - There can be many instances of one class

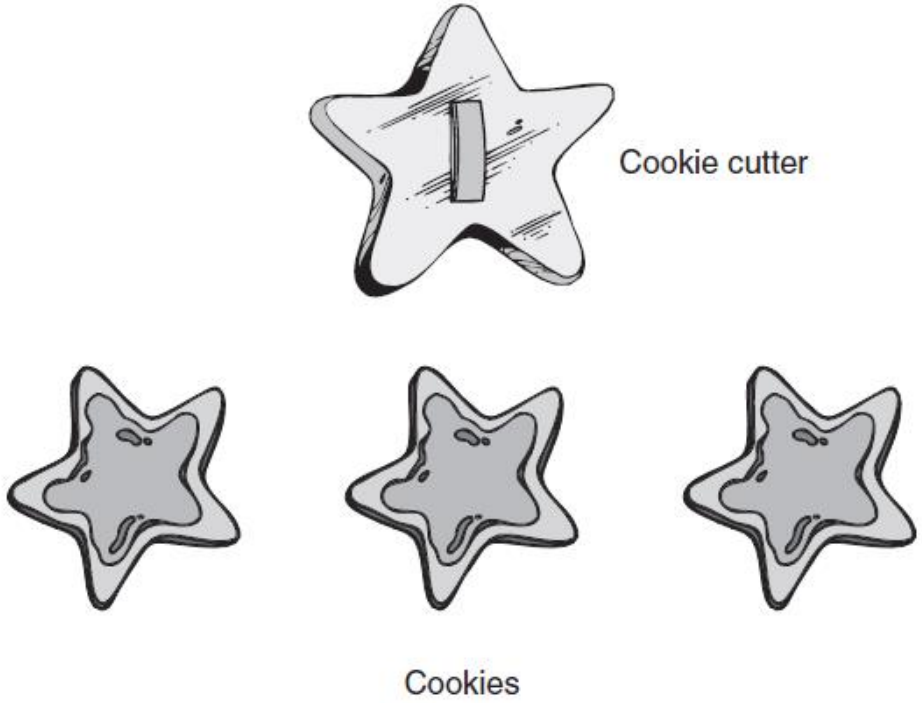
Classes (cont'd.)

Figure 10-3 A blueprint and houses built from the blueprint



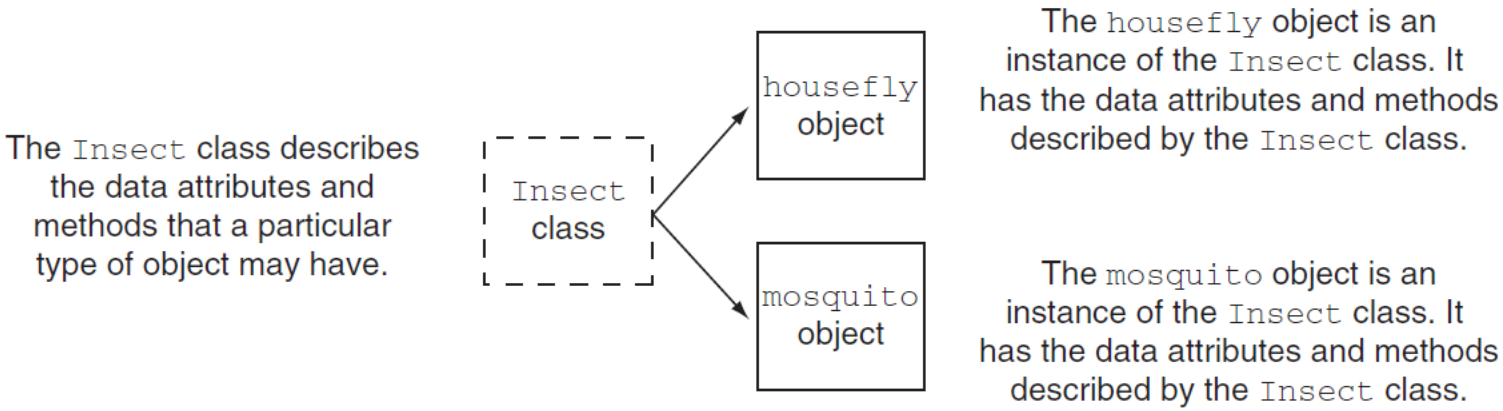
Classes (cont'd.)

Figure 10-4 The cookie cutter metaphor



Classes (cont'd.)

Figure 10-5 The `housefly` and `mosquito` objects are instances of the `Insect` class



Class Declarations

- Typically written in a file named “className.h”.
- Begins with keyword class followed by the name (identifier) of the new class type.
- Body of the class declaration is a block of code containing
  - declaration of data members (attributes)
  - method (function) prototypes
  - keywords public, protected, and private are used to control access to data members and methods
  - A semicolon must terminate the body of the class declaration. };

Class Implementation

- The class is typically written in a file named “className.cpp”
- File should `#include “className.h”`
- Provides the code to implement class methods.

Class Syntax

Syntax: Class Declaration
//filename:className.h

```
class className
{
     access modifier:
        declaration of attributes
     acccess modifier:
        declaration of methods
};
```

Syntax: Class Implementation
//filename: className.cpp

```
#include "className.h"
definitions of class methods
```

Example: class Declaration
//filename: Point.h

```
class Point
{
  private:
    double xCoord;
    double yCoord;

  public:
  Point(double x, double y);
};
```

Example: class Implementation
//filename: Point.cpp

```
#include "Point.h"

Point::Point (double x, double y)
{
  xCoord = x;
  YCoord = Y;
}
```

Class Methods

- Define the operations that can be performed on class objects.
  - A **constructor** is a special method that is executed when objects of t class type are declared (**instantiated**).
  - Constructors have the same name as the class.
  - A class may define multiple constructors to allow greater flexibility in creating objects.
    - The default constructor has no parameters.
    - Parameterized constructors provide initial values for data member

Using a Class

Usage:

```
#include "Point.h"
...
int main()
{
  Point p1(1.5, 2.7);
...
```

- Once a class is defined, you may use the class name as a type specifier.
  - You must include the class declaration (i.e. header file)
  - You must link to the class implementation (i.e. `.cpp` file)