SSC - Networks and Concurrency

Dr Shan He

School for Computational Science
University of Birmingham

Module 06-19321: SSC
Outline of Topics

About this module

Something about programming
Lectures and tutorials

- Three lectures per week
  - Monday 9:00am, Tuesday 9:00am and 15:00pm
  - New material: theory, practice and Java examples

- Tutorials:
  - No preallocated time/location and depend on progress
  - Interview questions, exam questions, and new trends in Java programming

- Lab sessions:
  - Thursday 11:00am-13:00pm
  - Friday 11:00am-13:00pm

- Please feel free to ask me questions:
  - My email: s.he@cs.bham.ac.uk
  - My office hour: Thursday 4:00pm
Continuous assessment

- 20% of your final mark.
- Three programming assignments:
  - Communications in Java: Issued in week 5, viva end of Week 7
  - Concurrency in Java: Issued in week 7, viva end of Week 8
  - Java web applications: Issued in week 8, viva start of Week 11
Aims of the module: Learning outcomes

- Explain and implement socket based communication
- Explain the concepts and issues of threads and synchronisation
- Implement, debug and test multithreaded programs with appropriate synchronisation
- Explain and evaluate web application architecture and technologies
- Explain the servlet model and structure
- Configure, install and use servlet based web applications, implementing correct servlet request and response code, including html form processing, session handling, database connectivity and security
What will be covered

- Communication & Networking (2 weeks)
  - Background
  - Java socket programming: TCP and JavaMail API
  - Java socket programming: UDP multicasting and Java NIO
  - Working with URL and HTML using Java

- Concurrency (3 weeks)
  - Background
  - Java multi-threaded programming and debugging
  - Patterns for concurrent systems
What will be covered

- Web applications (2 weeks)
  - Background
  - Java Servlet programming
  - Web application programming (Servlet, MySQL, JDBC and Node.js)
- Revision lessons: past exam questions
- Java interview questions
Something about programming
Programming learning cycle

- Step 1: Study new concepts;
- Step 2: Search, read and execute program examples;
- Step 3: Compare output to expectation;
- Step 4: Iterate until the two match;
- Step 5: Write your own programs following examples;
- Step 6: If not enough, go to Step 1.
How to learn programming

One interesting article
Object-oriented programming in Java

- **Object**: a software bundle of related state and behaviour, which consists of:
  - **Fields**: An object stores its state in fields (variables)
  - **Methods**: An object exposes its behaviour by methods, e.g., operate on an object’s internal state and communicate with other objects

- **Class**: a blueprint or prototype or template from which objects are created

- **Inheritance**: inherit commonly used state and behavior from other classes. Better ways for organising and structuring your software

- **Interface**: a contract between a class and the outside world.

- **Package**: a namespace that organizes a set of related classes and interfaces - similar to folders in your computer.
Q&A: Java OOP concepts

- What are the differences between class and object?
Q&A: Java OOP concepts

- A class
  - is a template for objects
  - defines object states: valid ranges of fields, default values of fields
  - describes object behavior by methods: define methods
  - can have sub-classes

- An object
  - is an instance of a class
  - have a lifespan but classes do not
  - cannot have sub-object
Q&A: Java OOP concepts

- List significant features of Object-oriented programming
Q&A: Java OOP concepts

- Q: List significant features of Object-oriented programming
- A:
  - **Encapsulation**: the ability to hide their internal data and behaviour. Each object provides a number of methods, which can be accessed by other objects and change its internal data.
  - **Inheritance**: the ability to acquire the fields and methods of another class, called base class. Inheritance provides re-usability of code and can be used to add additional features to an existing class, without modifying it.
  - **Polymorphism**: the ability to present the same interface for differing underlying data types.
  - **Abstraction**: the process of separating ideas from specific instances of those ideas at work, in other words, we will develop classes in terms of their own functionality, instead of their implementation details.
Q&A: Java OOP concepts

- List advantages of Object-oriented programming
Q&A: Java OOP concepts

▶ Q: List 4 significant features of Object-oriented programming
▶ A:
  ▶ **Simplicity**: able to reduce the software complexity by modelling the real world in a more natural but more complete fashion, e.g., the world can be modelled as classes of objects, and objects are associated with behaviours.
  ▶ **Modularity**: the ability to enforce logical boundaries between components, e.g., each object forms a separate entity whose internal workings are decoupled from other parts of the system, which leads to easy maintenance and modification.
  ▶ **Reusability**: the ability to create a new class that uses the features of an existing class without recoding those features.
  ▶ **Reliability**: able to build new behaviours based on existing reliable objects by adding new functionality to it.
Questions: Java OOP concepts

- Real-world objects contain ____ and ____
- A software object’s state is stored in ____
- A software object’s behavior is exposed through ____
- Hiding internal data from the outside world, and accessing it only through publicly exposed methods is known as data ____
- Common behavior can be defined in a ____ and inherited into a ____ using the ____ keyword.
- A collection of methods with no implementation is called an ____
- A namespace that organizes classes and interfaces by functionality is called a ____
Want more questions?

Top 50 OOP interview questions