1. Investigate the role played by an Operating System. Describe how an OS allows software to interact with hardware.

   a. An Operating System is a system which commands and operates the software installed on a computer and allows the user to interact with the hardware of the computer. The OS includes drivers, process management systems and other utility software which allows it to bridge the gap between the hardware of a device and the software running on it by providing a platform which has 'hooks' to the hardware components, so that software on the computer can use common methods to access the hardware components.

2. Compare and contrast Application Software and Utility Software.

   a. Application Software is software that a normal user operates and uses on a daily basis, such as document editing programs, email programs, etc. These types of programs are made with the intent of using the computer and its resources to create something or perform an action.

   b. Utility Software is software that is used by the operating system to keep the computer running in normal working order and are generally more concerned with the function of the device than its purpose. Utility software is software such as antivirus programs, filesystem navigators, and disk monitoring.

3. List and describe three utility software usually included in an OS

   a. A disk utility program, used for formatting, cleaning, partitioning and repairing storage drives.

   b. A file explorer, used for browsing files on a storage device, usually including options to delete, rename, duplicate and create files and folders.

   c. A GUI telemetry or monitoring program, so the user can see computer performance statistics such as memory usage, central processing unit load and temperature readings

4. Research one of the three main desktop Operating Systems. Find screenshots displaying key features of their major versions and collect into a timeline in PowerPoint.

MAC OS X

MAC OS X 10.0 CHEETAH

‣ Released on March 24, 2001 for a price of $129 US Dollars

‣ Designed and Made by Apple

‣ The first major release of Mac OS X

‣ Featured an application and file Dock, a Terminal, an Email client, Text editor, PDF Support, AppleScript and a new Mach 3.0 microkernel

‣ It also introduced protected memory, where if one application corrupted its memory, it would not affect other applications.
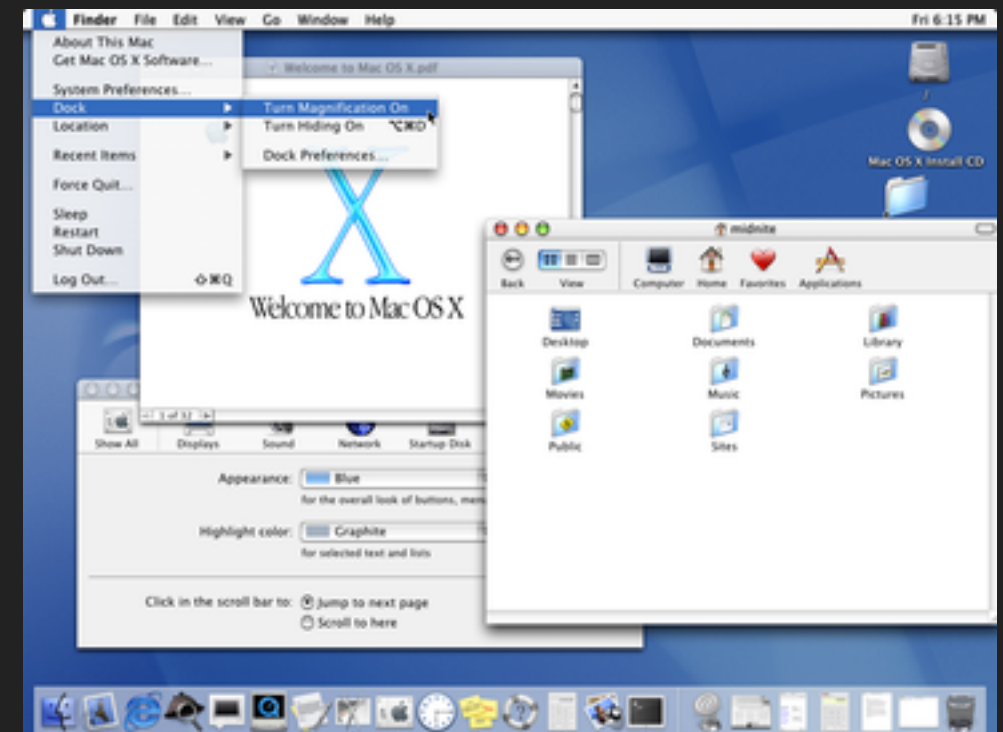
Mac OS X 10.0 Cheetah,

featuring the Dock, Finder, and Preview

MAC OS X 10.7 LION

▸ Released on July 20, 2011 for a price of $30 US Dollars on the Mac App Store

▸ Known as the iOS update, as it brought many good things and integrations from iOS to the Mac

▸ Main features include:

  ▸ AirDrop, Lion-to-Lion file sharing with no wireless access point required

  ▸ Launchpad, a place to show all of your applications in one spot, inspiration taken from iOS

  ▸ Mission Control, to see all of your application windows and manage them.

  ▸ Preview speed updates, and a new signature scanning feature to quickly place a vectorised version of your signature in documents by taking a photo of it.

Mac OS X 10.7 Lion,

featuring its new LaunchPad

MAC OS X 10.10 YOSEMITE

‣ Released on March 16, 2014 as a free upgrade via the App Store

‣ New features:

   ‣ Handoff, answer calls on your Mac, open your iOS browser session on your Mac

   ‣ A major overhaul of the user interface. Tonal and gradient graphics were replaced with flat design and blurred translucency effects, similar to the design introduced by iOS 7.

   ‣ Photos replaced iPhoto and Aperture, with heavy iOS integrations

Mac OS X 10.10 Yosemite,

featuring the new flat design and Handoff, with a call coming in from Notification Centre

MacOS 10.12 SIERRA

▸ Released on September 20, 2016 as a free upgrade via the App Store

▸ Was rebranded as macOS to suit the rest of Apple's operating systems, such as iOS, tvOS, watchOS

▸ New features include:

  ▸ Siri, who lives in notification centre and can search files, the web, or change some computer settings

  ▸ iCloud Drive, similar to Dropbox and Google Drive, also a new application to analyse where you can free up disk space.

  ▸ More continuity between iOS, including automatic unlock with an Apple Watch and a Universal Clipboard between iOS and macOS

Mac OS X 10.12 Sierra,

featuring Siri, a new disk space analyser and a Pages, Keynote and Numbers update

5. Compare and contrast the features of the three major desktop Operating Systems.

   a. Windows

      i. Good for businesses and users looking for compatibility, with thousands and thousands of programs designed for Windows.

      ii. Hundreds of thousands of businesses also use Windows systems and and for cross-system compatibility Windows is the best option.

      iii. Can be installed on virtually any machine, and only requires a licence key to work.

      iv. Windows is a very unstable operating system, and it is recommended by Microsoft, to re-install the entire system every 6 months.

      v. Highly susceptible to viruses.

   b. macOS

      i. Good for artists of any kind, including musical, photographical and designers. Many apps have been made by Apple for these purposes, and are designed solely with Mac in mind. Other products such as the Adobe Suite are also available for macOS.

      ii. Very stable and secure operating system.

      iii. macOS is also very good for people who are new to computers, as it has a very simple user interface.

      iv. Only runs on Macintosh computers, but comes with every Mac for free.

   c. Linux

      i. Completely open source and free, runs on any machine.

      ii. There is a version on Linux designed for virtually any task, whether it be running servers, gaming, etc. Also hard to pick a version of Linux if you are not very familiar with computers.

iii. Very stable operating system and hence very commonly used for web servers.

iv. Very light and streamlined system, does not come with lots of bulky addons, such as drivers, like macOS and Windows do

v. Not very user-friendly, and are not very good for people who are new to computers.

1. Replicate the steps of the recipe in pseudocode.

```
BEGIN makePizza

    SET ingredients to
array[flour,salt,yeast,tpaste,cheese,pepperoni,pepper,warmwater,oliveoil]

    SET Oven.Settings to [on,200,celsius,fanforced]
    SET mixingbowl to empty array
    SET board to choppingBoard

    GET ingredients[0-8]

    ADD ingredients[7-8] to mixingbowl
    ADD ingredients[0-2] to mixingbowl

    SET mixture to mixingbowl.Contents

    DO
       STIR mixture
    UNTIL mixture.Consistency is solid and thick

    SPRINKLE ingredient[0] on board
    TIP mixture onto board

    SET timer to new stopwatch
    DO
       KNEAD mixture
    UNTIL timer.Time > 300 seconds OR mixture.Consistency is smooth

    SHAPE mixture into flat disc

    DO
       COVER mixture with ingredients[3]
    UNTIL mixture is covered in ingredients[3]
    DO
       COVER mixture with ingredients[4]
    UNTIL mixture is covered in ingredients[4]
    FOR a = 1 to 20
       PLACE ingredients[5] on mixture
    NEXT
    DO
       SPRINKLE ingredients[6] on mixture
    UNTIL mixture is well seasoned

    SET sheet to new baking paper sheet
    SPRINKLE sheet with ingredients[0]
    PLACE mixture on sheet

    WHILE Oven.CurrentTemperature < 200

       PRINT "Heating..."

    END WHILE

    PLACE mixture in Oven

    SET timer2 to new stopwatch
    DO
       PRINT "Cooking... "
    UNTIL timer2.Time > 600 OR mixture.Colour is golden

    REMOVE mixture from Oven

END makePizza
```

2. Construct an algorithm that sorts packages in an assembly line between three locations based on the colour of the package . The program must also count the number of each

type of package that goes to each location and the average number of packages sent in a day.

```
BEGIN packageSorter

    SET redCount, greenCount, blueCount & workerCount to 0

    while unsorted packages exist

        DO
            roll assembly line conveyor forward
        UNTIL new package detected

        SET scannerColour to colour returned by ColourSensor
        SET conveyorUsed to 0

        CASE scannerColour
            CASEWHERE red:
                move package to conveyor 1
                redCount = redCount + 1
                conveyorUsed = 1
            CASEWHERE green:
                move package to conveyor 2
                greenCount = greenCount + 1
                conveyorUsed = 2
            CASEWHERE blue:
                move package to conveyor 3
                blueCount = blueCount + 1
                conveyorUsed = 3
            CASEWHERE default:
                move package to conveyor 4 // Worker Inspection
                workerCount = workerCount + 1
                conveyorUsed = 4
        ENDCASE

        roll conveyor[conveyorUsed] forward

    END while

    SET todayCount to redCount + blueCount + greenCount + workerCount
    SET previousCounts to contents of data file in integer array
    add todayCount to end of previousCounts
    write previous counts to data file

    SET averageCount to 0

    FOR a = 1 to previousCounts(count)

        averageCount = averageCount + 1

    NEXT

    averageCount = averageCount / previousCounts(count)

    PRINT "Red Packages Sorted: " + redCount
    PRINT "Blue Packages Sorted: " + blueCount
    PRINT "Green Packages Sorted: " + greenCount
    PRINT "Packages sent to Workers: " + workerCount

    PRINT "Packages sent today: " + todayCount
    PRINT "Average Packages sent recently: " + averageCount

END packageSorter
```

3. Construct an algorithm that puts random numbers in a 10 by 10 grid (like a spreadsheet). You can assume the sub routine get_random() is written elsewhere.

```
BEGIN fillGrid

  grid = 10 by 10 array

  FOR x = 1 to 10

    FOR y = 1 to 10

      SET targetCell to grid(x,y)

      SET targetCell(contents) to get_random()

    NEXT

  NEXT

END fillGrid
```

```
START
checkHits
```

```
Are there
aliens in
arrAliens?
```

NO →
```
PLAYER
WINS/
NEXT WAVE
```

YES →
```
Set current
alien to next
alien in
arrAliens
```

```
Is the current
alien intersecting
with any player
bullets?
```

YES →
```
DELETE
ALIEN
```
↓
```
PLAY HIT
SOUND
```
↓
```
INCREASE
PLAYER
SCORE
```

NO ↓

```
Have all aliens
in arrAliens
been checked?
```

NO →

YES →

```
Are any alien
bullets
intersecting with
the ship?
```

YES ↑
```
DEDUCT
HEALTH POINT
FROM PLAYER
```
→
```
PLAY SHIP
HIT SOUND
```
→
```
Does the player
have less than
one health point?
```

YES ↓
```
PLAYER
LOSES/
GAME OVER
```

NO →

```
Is the ship
intersecting with
the powerup?
```

YES ↑
```
GIVE
POWERUP
TO PLAYER
```
→
```
HIDE
POWERUP
```

NO →

NO →
```
END
```

```
START
moveBullets
```

Are there
bullets in
arrBullets ?

—NO→ END

YES

Set current
bullet to next
bullet in
arrBullets

Is the current
bullet out of
the screen?

—YES→ DELETE
BULLET

NO

Who was the
current bullet
fired by?

ALIEN → MOVE
BULLET
DOWN

SHIP → MOVE
BULLET UP

Have all bullets
in arrBullets
been checked?

NO

YES

END

```
START TicTacToe → Who is playing first?
```

Who is playing first? —AI→ Play center
Who is playing first? —User→ Did the user play in the center?

Play center → Wait for user to select box

Did the user play in the center? —NO→ Play center
Did the user play in the center? —YES→ Play in any free corner

Play in any free corner → Wait for user to select box

Wait for user to select box → Can the user win on the next round?

Can the user win on the next round? —NO→ Play in any free corner
Can the user win on the next round? —YES→ Play where the user would have to to win

Play in any free corner → Wait for user to select box
Play where the user would have to to win → Wait for user to select box

Wait for user to select box → Can the computer win this round?

Can the computer win this round? —NO→ Can the user win on the next round?
Can the computer win this round? —NO→ Does the user have three in a row?
Can the computer win this round? —YES→ Play to complete the line

Does the user have three in a row? —YES→ USER WINS

Play to complete the line → COMPUTER WINS

Can the user win on the next round? —NO→ Is there a free corner?
Can the user win on the next round? —YES→ Play where the user would have to to win

Is there a free corner? —NO→ Play any side
Is there a free corner? —YES→ Play in any free corner

Play any side → Is the board full?
Play in any free corner → Is the board full?
Play where the user would have to to win → Is the board full?

Is the board full? —YES→ TIE
Is the board full? —NO→ Does the user have three in a row?

Wait for user to select box → Does the user have three in a row?

Does the user have three in a row? —YES→ USER WINS
Does the user have three in a row? —NO→ TIE