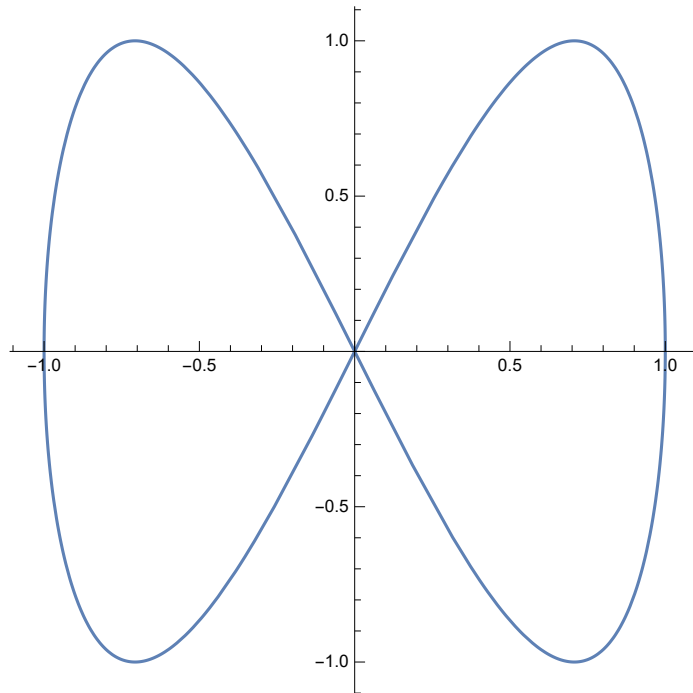
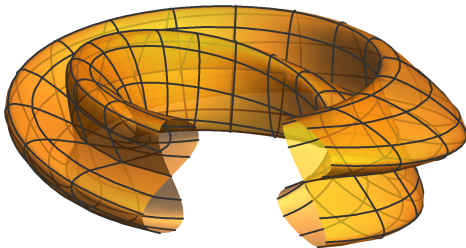


Pensieve header: Class of March 7: Packaging.

```
ParametricPlot[{Sin[t], Sin[2 t]}, {t, 0, 2 Pi}]
```



```
ParametricPlot3D[
  r = {Cos[2 t], Sin[2 t], 0}; n = {0, 0, 1};
  3 r + (Cos[t] r - Sin[t] n) Sin[2 s] + (Sin[t] r + Cos[t] n) Sin[s],
  {t, 0, 7 Pi/8}, {s, 0, 2 Pi},
  PlotStyle -> Opacity[.7], Axes -> None, Boxed -> False, Background -> White,
  ViewPoint -> {4.68705, -1.12888, 0.870159},
  ViewVertical -> {-0.194434, 0.141663, 3.30223}
]
```



From <http://www.math.toronto.edu/drorbn/classes/16-1750-ShamelessMathematica/About.html>: **Possible Topics** (in no particular order). Whatever you may suggest, and the Fibonacci numbers; the Jones polynomial; a more efficient Jones-

algorithm; a riddle on spheres; Khovanov homology; Γ -calculus; the Hopf fibration; Hilbert's 13th problem; non-commutative Gaussian elimination; free Lie algebras; the Baker-Campbell-Hausdorff formula; wacky numbers; an order 4 torus; the Schwarz Lantern; knot colourings; the Temperley-Lieb pairing; the dodecahedral link; sound experiments; barycentric subdivisions; a Peano curve; braid closures and Vogel's algorithm; the insolubility of the quintic; phase portraits.

Package for use by others

1. Make into a "package".
2. Save as a package (.m) file.
3. Do not enforce variable names.
4. Document.
5. Allow "mod p".
6. Allow "verbose".
7. Print error messages, as needed.

? \$Context

\$Context is a global variable that gives the current context. >>

? \$ContextPath

\$ContextPath is a global variable that gives a list of contexts, after \$Context, to search in trying to find a symbol that has been entered. >>

? BeginPackage

BeginPackage["context"] makes context` and System` the only active contexts.
BeginPackage["context", {"need1", "need2", ...}] calls Needs on the needi. >>

? EndPackage

EndPackage[] restores \$Context and \$ContextPath to their values before the preceding BeginPackage, and prepends the current context to the list \$ContextPath. >>

? Begin

Begin["context"] resets the current context. >>

? End

End[] returns the present context, and reverts to the previous one. >>

```

BeginPackage["SqSq`"];
SqSq;
Begin["`privet`"];
SqSq[z_] := (sq = z^2; sq^2);
End[];
EndPackage[];

```

? ::

symbol::tag is a name for a message. >>

? Modulus

Modulus $\rightarrow n$ is an option that can be given in certain algebraic functions to specify that integers should be treated modulo n . >>

? Options

Options[*symbol*] gives the list of default options assigned to a symbol.
 Options[*expr*] gives the options explicitly specified in a particular expression such as a graphics object.
 Options[*stream*] or Options["*sname*"] gives options associated with a particular stream.
 Options[*object*] gives options associated with an external object such as a NotebookObject.
 Options[*obj, name*] gives the setting for the option *name*.
 Options[*obj, {name₁, name₂, ...}*] gives a list of the settings for the options *name_i*. >>

? Message

Message[*symbol::tag*] prints the message *symbol::tag* unless it has been switched off.
 Message[*symbol::tag, e₁, e₂, ...*] prints a message, inserting the values of the *e_i* as needed. >>

? Off

Off[*symbol::tag*] switches off a message, so that it is no longer printed.
 Off["*name*"] switches off a named group of messages.
 Off[*s*] switches off tracing messages associated with the symbol *s*.
 Off[*m₁, m₂, ...*] switches off several messages or message groups. >>

? On

On[*symbol::tag*] switches on a message, so that it can be printed.
 On["*name*"] switches on a named group of messages.
 On[*s*] switches on tracing for the symbol *s*.
 On[*m₁, m₂, ...*] switches on several messages or message groups. >>

? Abort

Abort[] generates an interrupt to abort a computation. >>

? Return

Return[*expr*] returns the value *expr* from a function.

Return[] returns the value Null. >>