Object-Oriented Programming Concepts

Session: 10

Generics

Objectives

- Explain Generics
- List ways of implementing Generics
- Explain Template Function
- Explain Template Class
- List advantages and disadvantages of Generics

Generics 1-2

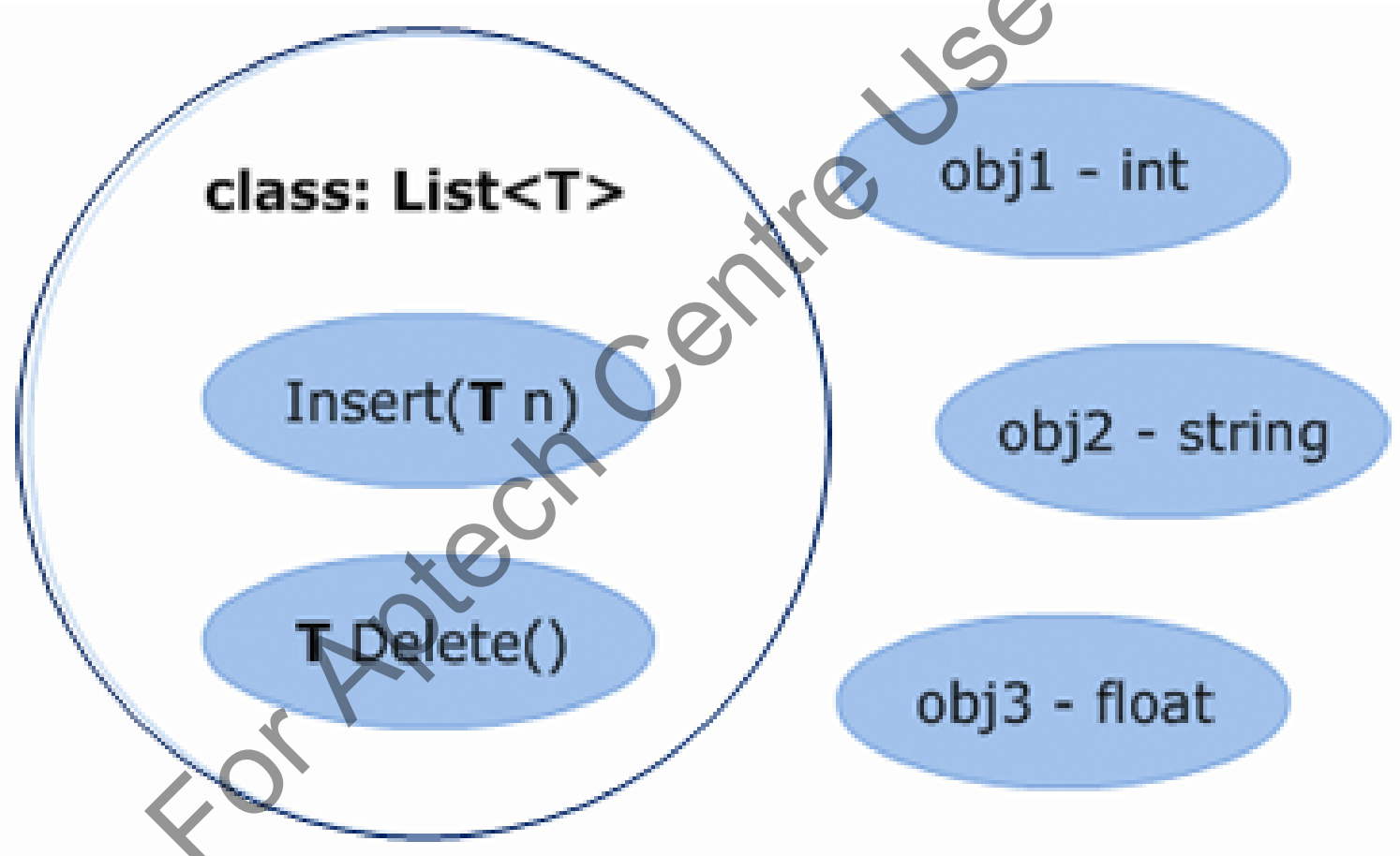
Generics allow creating data members and functions, whose the type is not specified at compile time.

The type of the value of a generic function or variable is decided later based on value supplied by user.

It allows the programmer to define the behavior of the class without knowing the data type.

Generics 2-2

- The figure shows an example of generics.



Template Functions

- It does not have a specific type attached to the parameters or even the return type.
- The type for the parameters is decided based on the value passed by the user at runtime.

Syntax

```
<access-modifier> <T> <method-name>
(<T> <parameter-name>,..)
{
      // processing statements
      <return-value (or expression)>
}
```

Template Class

Template class consists of a generic type declaration.

The generic type introduces a type variable named 'T' which can be used anywhere inside the class.

Syntax

```
<access-modifier> class <class-name> <T>
{
      <data-members>

      <methods>
}
```

Advantages of Generics 1-2

- Generics allow a programmer to create classes with flexibility in applying the data type.
- The data type is decided when the object is created and not when the class is created.
- Generics allow overloading of template functions with parameters of unspecified types.
- The main benefit of generics is late binding of the types.

Advantages of Generics 2-2

- The most common use of generics is the creation of typed Collections such as List, Stack, and Array
- The compiler determines the type associated with the template that can perform all the required functions

Disadvantages of Generics

- Some compilers have poor support for templates.
- Some compilers lack in clear description when they detect a template error.
- The compiler usually generates additional code for each template type.
- Nested templates are not supported by all compilers.
- The use of ‘less-than’ and ‘greater-than’ signs as delimiters causes problems.

Summary 1-2

- Generics provides a way of creating general purpose tools for applying them to specific situations by making relevant changes.
- Generics is also known as templates in C++.
- A template function is a method that does not have a specific type attached to the parameters or even the return type.

Summary 2-2

- A template class introduces a type variable 'T' which can be used as a type with the data members and methods of a class.
- Generics allow overloading of template functions with parameters of unspecified types.
- The language C++ uses the keyword ‘template’ to indicate that a class is a template class using generics.
- The most common use of generics is the creation of typed Collections such as List, Stack, and Array.