***NAME : Himanshu Dixit***

***ENROLL NO. : 21103262***

***BATCH : B10***

***Software Development fundamentals-2 [EVEN 2022]***

***Tutorial Sheet -7 (Week 7)***

***1.*** *Identify the relationship(s); analyze the sequence of constructor and destructor invocation; and obtain the output-*

|  |  |
| --- | --- |
| *#include<iostream>*  *using namespace std;*  *class ABC*  *{*  *int x; public:*  *ABC() { cout<<"1\t"; }*  *~ABC() { cout<<"2\t"; }*  *};*  *class KLM*  *{*  *int y;*  *ABC O1;*  *public:*  *KLM() { cout<<"3\t"; }*  *~KLM() { cout<<"4\t"; }*  *};* | *class XYZ*  *{*  *int y;*  *KLM O2;*  *ABC O3;*  *public:*  *XYZ() { cout<<"5\t"; }*  *~XYZ() { cout<<"6\t"; }*  *};*  *int main()*  *{*  *XYZ \*O4=new XYZ();*  *delete(O4);*  *return 0; }* |

***Solution:***

Output: 1 3 1 5 6 2 4 2

Relationship between ABC and KLM is composition.KLM is composed by ABC

Relationship between ABC and KLM and XYZ is composition.XYZ is composed by ABC and KLM

***2.*** *Identify the relationship(s); analyze the sequence of constructor and destructor invocation; and obtain the output-*

|  |  |
| --- | --- |
| *#include<iostream>*  *using namespace std;*  *class ABC*  *{*  *int x; public:*  *ABC() { cout<<"1\t"; }*  *~ABC() { cout<<"2\t"; }* | *int main()*  *{*  *KLM \*O4=new KLM();*  *ABC a1;*  *delete(O4);*  *return 0;* |

|  |  |
| --- | --- |
| *};*  *class KLM*  *{*  *int y;*  *ABC \*O1;*  *public:*  *KLM() { cout<<"3\t"; }*  *~KLM() { cout<<"4\t"; }*  *};* | *}* |

***Solution:***

Output: 3 1 4 2

Relationship between ABC and KLM is association.KLM is associated by ABC

***3.*** *Identify the relationship(s); analyze the sequence of constructor and destructor invocation; and obtain the output-*

|  |  |
| --- | --- |
| *#include<iostream>*  *using namespace std;*  *class ABC*  *{*  *int x; public:*  *ABC() { cout<<"1\t"; }*  *~ABC() { cout<<"2\t"; }*  *};*  *class KLM*  *{*  *int y;*  *ABC \*O1;*  *public:* | *KLM() { cout<<"3\t";*  *O1= new ABC;*  *}*  *~KLM() { cout<<"4\t"; }*  *};*  *int main()*  *{*  *KLM \*O4=new KLM;*  *ABC a1;*  *delete(O4);*  *return 0; }* |

***Solution:***

Output: 3 1 1 4 2

Relationship between ABC and KLM is aggregation.KLM is aggregrated by ABC

***4.*** *Identify the relationship(s); analyze the sequence of constructor and destructor invocation; and obtain the output-*

|  |  |
| --- | --- |
| *#include<iostream>*  *using namespace std;*  *class ABC*  *{*  *int x; public:*  *ABC() { cout<<"1\t"; }*  *~ABC() { cout<<"2\t"; }*  *};*  *class KLM*  *{*  *int y; ABC \*O1;*  *public:* | *class XYZ*  *{*  *int y; ABC O2; KLM O3;*  *public:*  *XYZ() { cout<<"5\t"; }*  *~XYZ() { cout<<"6\t"; }*  *};*  *int main()*  *{ XYZ O4, \*O5;*  *KLM \*O6=new KLM;*  *O5 = new XYZ;*  *delete(O6);*  *return 0; }* |

|  |  |
| --- | --- |
| *KLM() { cout<<"3\t"; O1 = new ABC;}*  *~KLM() { cout<<"4\t";}*  *};* |  |

***Solution:***

Output: 1 3 1 5 3 1 1 3 1 5 4 6 4 2

Relationship between ABC and KLM is aggregation.KLM is aggregrated from ABC

Relationship between ABC and KLM and XYZ is composition.XYZ is composed from ABC and KLM

***5.*** *Create a domain model for a hospital to show and explain hospital structure, staff, relationships with patients, and patient treatment terminology. The domain model for the Hospital Management System is represented by several class diagrams. Consider the following relationships in the domain model.*

*1. Ward is a division of a hospital or a suite of rooms shared by patients who need a similar kind of care. In a hospital, there are a number of wards, each of which may be empty or have on it one or more patients. Each ward has a unique name.*

*2. The doctors in the hospital are organised into teams (also called firms). Each team has a unique name or code (e.g. Orthopaedics or Pediatrics) and is headed by a consultant doctor or an attending physician.*

***Solution:***





