|  |  |  |
| --- | --- | --- |
|  | **SOFTWARE ENGINEERING**  **SWD392 – Progress Test 3** | |
| **Testing date**:………………… **Duration**: 45’  **Full Name:** ……………………………………….  **Roll Number:** ……………………………………… | **Testing score (/10):**………. |
| **CODE: SWD392.PT3.01** |

# Testing notes

* Fill your answers into the ANS column.
* You are allowed to use any communicating devices (laptop, mobile phone, etc…) during the test.
* You are not allowed to discuss with classmates.

# Testing questions & answers

|  |  |  |
| --- | --- | --- |
| **#** | **Question & Answer Group** | **ANS** |
| 1 | What are elements of a design pattern?  (a) Problem.  (b) Name, Solution and Consequences.  (c) Purpose.  (d) a & b. | d |
| 2 | What are pros of design pattern?  (a) Add consistency to designs by solving similar problems the same way.  (b) Add clarity to design.  (c) Improve time to solution by providing templates.  (d) All above answers. | d |
| 3 | What are cons of design pattern? (Multi-choice)  (a) Developers must take time draw class diagrams.  (b) Some patterns cause negative consequences in some context.  (c) Consequences are subjective depending on concrete scenarios.  (d) Some patterns cause longer code and shorter code simultaneously. | b, c |
| 4 | What are groups or (families) of design patterns? (Multi-choice)  (a) Creational.  (b) Relational.  (c) Behavioral.  (d) Structural. | a, c, d |

|  |  |  |
| --- | --- | --- |
| 5 | What is general purpose of the Creational design patterns?  (a) Provide various object creation mechanisms.  (b) A cyclic approach to developing class diagram  (c) An approach to design structure in software architecture.  (d) Provide various object creation mechanisms, which increase flexibility and reuse of existing code. | d |
| 6 | What is general purpose of the Structural design patterns? (a) Explain how to assemble objects and classes into larger structures, while keeping these structures flexible and efficient. (b) Describe class in structure of code. (c) Describe class diagram in style of structure for architecture design. (d) Assemble class into a class diagram. | a |
| 7 | What is general purpose of the Behavioral design patterns? (a) Organize processing mechanism between objects.  (b) Create form for classes in an algorithm. (c) Define dynamic processing between classes. (d) Organize algorithms and the assignment of responsibilities between objects. | d |
| 8 | What is design pattern? (a) Set of feature of Object-Oriented programming languages. (b) Programming technique. (c) Naming method for code re-factoring. (d) Set of templates to do software detail design. | d |
| 9 | How many design patterns are in the standard document "Gang of Four"?  (a) 10.  (b) 23.  (c) 4.  (d) 14. | b |
| 10 | Which design pattern is the below source code?  (a) Factory Method.  (b) State.  (c) Adapter.  (d) Facade.  **class B {**  **private:**  **A \*a;**  **public:**  **B() { a = new A; }**  **string Action() {**  **string result = a->Action(' ', ' ', "");**  **return result;**  **}**  **};**  **class A {**  **public:**  **string Action(char, char, string) { return "A"; }**  **};**  **void ClientCode() {**  **B \*b = new B;**  **print(b.Action());**  **}** | d |
| 11 | Which design pattern is the below source code?  (a) Decorator.  (b) Builder.  (c) Abstract Factory.  (d) Factory Method.  **class A {**  **private:**  **B\* b;**  **public:**  **A() { b = new B(); }**  **void CreateA() { b.list.add("A"); }**  **void CreateB() { b.list.add("B"); }**  **void CreateC() { b.list.add("C"); }**  **};**  **class B {**  **private:**  **List list;**  **public:**  **void ShowList() { ... }**  **};**  **void ClientCode() {**  **A \*a = new A;**  **a.CreateA();**  **a.CreateB();**  **a.CreateC();**  **}** | b |
| 12 | Which design pattern is the below source code?  (a) Decorator.  (b) Builder.  (c) Abstract Factory.  (d) Factory Method.  **class A {**  **public:**  **virtual ~A() {}**  **virtual string Operation() = 0;**  **};**  **class ConcreteA1 : A {**  **public:**  **string Operation() { return "Result of the ConcreteA1"; }**  **};**  **class ConcreteA2 : A {**  **public:**  **string Operation() { return "Result of the ConcreteA2"; }**  **};**  **class Creator {**  **public:**  **virtual ~Creator(){};**  **virtual A\* Produce() = 0;**  **string SomeOperation() {**  **A\* object = this->Produce();**  **string result = object->Operation();**  **return result;**  **}**  **};**  **class ConcreteCreator1 : Creator {**  **public:**  **A\* Produce() { return new ConcreteA1(); }**  **};**  **class ConcreteCreator2 : Creator {**  **public:**  **A\* Produce() { return new ConcreteA2(); }**  **};**  **void ClientCode() {**  **Creator creator;**  **print(creator.SomeOperation());**  **}** | d |
| 13 | Which design pattern is the below source code?  (a) Observer.  (b) State.  (c) Abstract Factory.  (d) Decorator.  **class ICenter {**  **public:**  **virtual void Attach(B b) = 0;**  **virtual void Notify() = 0;**  **};**  **class Center : ICenter {**  **private:**  **string m\_message;**  **List viewerList;**  **public:**  **void Attach(B b) { viewerList.add(b); }**  **void Notify() {**  **for (each b in viewerList) { B.Update(message); }**  **}**  **void ReceiveMessage(string message) {**  **m\_message = message;**  **Notify();**  **}**  **};**  **class B {**  **public:**  **virtual ~B() {};**  **virtual void Update(string message) = 0;**  **};**  **class ConcreteB : B {**  **private:**  **string inputMessage;**  **Center m\_center;**  **public:**  **ConcreteB(Center center) { m\_center.Attach(this); }**  **void Update(string message) {**  **inputMessage = message;**  **print("Processing for input message ...");**  **}**  **};**  **void ClientCode() {**  **Center \*center = new Center;**  **ConcreteB \*obj1 = new ConcreteB(center);**  **ConcreteB \*obj2 = new ConcreteB(center);**  **ConcreteB \*obj3 = new ConcreteB(center);**  **ConcreteB \*obj4 = new ConcreteB(center);**    **center->ReceiveMessage("Progress Test");**  **observer3->RemoveMeFromTheList();**  **}** | a |
| 14 | Which design pattern is the below source code?  (a) State.  (b) Builder.  (c) Abstract Factory.  (d) Facade.  **class Context {**  **private:**  **A \*m\_a;**  **public:**  **Context(A \*a) { ChangeTo(a); }**  **void ChangeTo(A \*a) {**  **if (m\_a != nullptr)**  **delete m\_a;**  **m\_a = a;**  **m\_a->setContext(this);**  **}**  **void Action1() { m\_a->Processing1(); }**  **void Action2() { m\_a->Processing2(); }**  **};**  **class A {**  **protected:**  **Context \*m\_context;**  **public:**  **void setContext(Context \*context) { m\_context = context; }**  **virtual void Processing1() = 0;**  **virtual void Processing2() = 0;**  **};**  **class ConcreteA : A {**  **public:**  **void Processing1() {**  **print("ConcreteA handles action1, then change a to ConcreteB");**  **m\_context->ChangeTo(new ConcreteB);**  **}**  **void Processing2() { print("ConcreteA handles action2"); }**  **};**  **class ConcreteB : A {**  **public:**  **void Processing1() { print("ConcreteB handles action1"); }**  **void Processing2() {**  **print("ConcreteB handles action2, then change a to ConcreteB");**  **m\_context->ChangeTo(new ConcreteA);**  **}**  **};**  **void ClientCode() {**  **Context \*context = new Context(new ConcreteA);**  **context->Action1();**  **context->Action2();**  **}** | a |
| 15 | Which design pattern is the below source code?  (a) Adapter.  (b) Facade.  (c) Builder.  (d) Strategy.  **class A**  **{**  **MathLib math;**  **PhysicsLib physics;**  **ChemistryLib chemis;**  **BiologyLib bio;**    **public:**  **void Calculate(int m, int p, int c, int b) {**  **mm = math.GetMath(m);**  **pp = physics.GetPhysics(p);**  **cc = chemis.GetChemistry(c);**  **bb = bio.GetBiology(b);**  **result = expression(mm, pp, cc, bb);**  **}**  **}**  **void Client() {**  **A student = new A();**  **student.Calculate(9, 10, 9, 9);**  **}** | b |

|  |  |  |
| --- | --- | --- |
| 16 | Which design pattern in the below Class diagram?  (a) Adapter.  (b) Abstract Factory.  (c) Facade.  (d) Strategy.  C:\Users\PCLTC\Desktop\SWD392 - Design patterns - Facade.jpg | c |
| 17 | Which design pattern in the below Class diagram?  (a) Facade.  (b) Factory Method.  (c) Singleton.  (d) Adapter.  C:\Users\PCLTC\Desktop\SWD392 - Design patterns-Adapter.jpg | d |
| 18 | Which design pattern in the below Class diagram?  (a) Strategy.  (b) Abstract Factory.  (c) Factory Method.  (d) Decorator.  C:\Users\PCLTC\Desktop\SWD392 - Design patterns-Abstract Factory.jpg | c |
| 19 | Which design pattern in the below Class diagram?  (a) Builder.  (b) Adapter.  (c) Decorator.  (d) Observer.  C:\Users\PCLTC\Desktop\SWD392 - Design patterns-Builder.jpg | a |
| 20 | Which design pattern in the below Class diagram?  (a) Facade.  (b) Observer.  (c) State.  (d) Builder.    C:\Users\PCLTC\Desktop\SWD392 - Design patterns-State.jpg | c |

**-- THE END --**