Exam1

● Graded

Student

Asrar Syed

Total Points

91.5 / 100 pts

Question 1

MC1 **3** / 3 pts

✔ **– 0 pts** Correct

– 3 pts Incorrect

Question 2

MC2 **3** / 3 pts

✔ **– 0 pts** Correct

– 3 pts Click here to replace this description.

Question 3

MS3-4 **6** / 6 pts

✔ **– 0 pts** Correct

– 3 pts Click here to replace this description.

Question 4

MS5-8 **12** / 12 pts

✔ **– 0 pts** Correct

– 3 pts Click here to replace this description.

– 6 pts Click here to replace this description.

Question 5

MC9 **3** / 3 pts

✔ **– 0 pts** Correct

– 3 pts Click here to replace this description.

Question 6

MC10 **3** / 3 pts

✔ **– 0 pts** Correct

– 3 pts Click here to replace this description.

Question 7

MC11 **3** / 3 pts

✔ **– 0 pts** Correct

– 3 pts Click here to replace this description.

Question 8

MS12-15 **9** / 12 pts

– 0 pts Correct

✔ **– 3 pts** Click here to replace this description.

– 6 pts Click here to replace this description.

– 9 pts Click here to replace this description.

Correct Answer - Analysis, Code Generation, Design, Documentation

Question 9

WR16 **5** / 5 pts

✔ **– 0 pts** Correct

– 2.5 pts Click here to replace this description.

– 5 pts Click here to replace this description.

– 1 pt Click here to replace this description.

Question 10

WR17 **5** / 5 pts

✔ **– 0 pts** Correct

– 1 pt Click here to replace this description.

– 2 pts Click here to replace this description.

Question 11

WR18 **2.5** / 5 pts

– 0 pts Correct

– 1 pt Click here to replace this description.

✔ **– 2.5 pts** Click here to replace this description.

– 5 pts Click here to replace this description.

– 0.5 pts Click here to replace this description.

Interface class cannot have concrete code (methods)
Abstract class can have concrete and abstract (methods)

Question 12

WR19 7 / 10 pts

– **0 pts** Correct

– **2 pts** Click here to replace this description.

✔ – **3 pts** Click here to replace this description.

– **4 pts** Click here to replace this description.

– **10 pts** Click here to replace this description.

Correct Answer :
Instead of using Iworker abstract class, remove the common methods into its own Interface class, then create two sub Interface classes, one for Salary Worker and one for Contract Worker . The sub Interface class have unique methods only for their respected implements.

Question 13

WR20 **20** / 20 pts

13.1 **WR20a** **10** / 10 pts

✔ – **0 pts** Correct

– **2 pts** Click here to replace this description.

– **1 pt** Click here to replace this description.

– **5 pts** Click here to replace this description.

– **8 pts** Click here to replace this description.

– **10 pts** Click here to replace this description.

13.2 **WR20b** **10** / 10 pts

✔ – **0 pts** Correct

– **10 pts** Click here to replace this description.

– **9 pts** Click here to replace this description.

– **1 pt** Click here to replace this description.

– **5 pts** Click here to replace this description.

– **0.5 pts** Click here to replace this description.

Question 14

WR21 **10** / 10 pts

✔ – **0 pts** Correct

– **2 pts** Click here to replace this description.

– **3 pts** Click here to replace this description.

PRINT NAME IN ALL CAPS: ASRAR SYED

GSU Panther ID: 002679083

Each multiple-choice answer or multi-select answer is valued at 3pts. **(total 45pts)** Write the capital letter (A, B, C, D) in the blank beside the question number in multiple-choice. ***Circled answers only will not be graded***, i.e., zero points if nothing is written in blank next to a question number. In multi-select (MS), place an '**X**' beside your choices. **Do not give lightly marked choices to confuse the grader**. Wrongly marked MS choices will be given negative points.

D 1. An OOP model for inheritance is what from the list below?
A) scaffolded
B) linear
C) circular
D) multi-level

B 2. An essential feature of OOP inheritance is what from the list below?
A) code insecurity
B) code reusability
C) code decoupling
D) code transparency

3-4MS Choose **2** fundamental features that OOP inheritance enables. (Mark a dark, clear '**X**' for choices)
_____ API exposure
X Encapsulation
_____ Virtual instance variables
X Polymorphism
_____ Integrated interfaces

5-8MS Choose **4** phases of the software development lifecycle (SDLC). (Mark a dark, clear '**X**' for choices)
X Requirements
_____ Interface expansion
X Design
X Implementation
X Evolution (upgrades)
_____ Documentation
_____ Code Optimizations

D 9. An important development technique in one of the 'early' SDLC phases is what from the list below?
A) Security testing
B) Code reduction
C) Open source
D) Prototyping

B 10. In the _____ phase of SDLC, the finished software is made available to users.
A) Implementation
B) Deployment
C) Upgrade
D) Planning

_____ 11. C is an ethics issue a software company faces in making tradeoffs between project schedules, project costs, and software quality.

A) Reducing quality assurance measurement
B) Enhanced security
C) Pressure to reduce costs
D) Ignoring open-source options

12-15MS Select **4** areas of software development where UML 'centric' (central connected) . (Mark a dark, clear '**X**' for choices)

X Analysis
_____ Agile Sprint cycles
_____ Code Generation
_____ Hiding defects
X Design
X Interface defects
X Documentation
_____ Manufacturer configuration

16) The Agile Manifesto has 4 key values, list **only two**. **(5pts)**

- working software over comprehensive documentation
- Customer collaboration over contract negotiation

17) Give two disadvantages of using a 'Waterfall' model in software development. **(5pts)**

- It is not suitable for complex projects as the risk factor is higher i.e. more errors or bugs that can't be caught because testing is later
- It is also expensive to go back to the design phase

18) Describe the one, main difference between an OOP abstract class and an interface class. **(5pts)**

abstract classes don't require static variable unlike interface classess. It can be used anywhere and it acts as an ancestor for classes that extend them.

19) A given example using java class files is violating S.O.L.I.D. principles of OOP (repeated interface code).

Interface Segregation

Problematic code:

```
public class SalaryWorker implements IWorker {
    private String ID, Name, Email;
    private double MonthSalary, MonthBenefit;
    private double HrRate, HrMonthly;
    public String getID(){ return ID; }
    public void setID(String _ID){ }
    public String getName(){ return Name; }
    public void setName(String _name){ }
    public String getEmail(){ return Email; }
    public void setEmail(String _email){ }
    public double getMonthSalary(){ return MonthSal; }
    public void setMonthSalary(double _salary){ }
    public double setMonthBenefit(double _Mon ){ }
    public void setMonthBenefit(){ return MonthBenefit; }
    public double getHourRate(){ return HrRate }
    public void setHourRate(double _hr_Rate){ }
    public double getHoursMonthly(){ return HrMonthly; }
    public void setHoursMonthly(double _Hours_Month){ }
    public double CalculatePay(){
        return getMonthSalary()*getMonthBenefit();
    }
}
```

```
public class ContractWorker implements IWorker {
    private String ID, Name, Email;
    private double MonthSalary, MonthBenefit;
    private double HrRate, HrMonthly;
    public String getID(){ return ID; }
    public void setID(String _ID){ }
    public String getName(){ return Name; }
    public void setName(String _name){ }
    public String getEmail(){ return Email; }
    public void setEmail(String _email){ }
    public double getMonthSalary(){ return MonthSal; }
    public void setMonthSalary(double _salary){ }
    public double setMonthBenefit(double _Mon ){ }
    public void setMonthBenefit(){ return MonthBenefit; }
    public double getHourRate(){ return HrRate }
    public void setHourRate(double _hr_Rate){ }
    public double getHoursMonthly(){ return HrMonthly; }
    public void setHoursMonthly(double _Hours_Month){ }
    public double CalculatePay(){
        return getHourRate()*getHoursMonthly();
    }
}
```

Describe (not using java code) a solution using the S.O.L.I.D. principle of 'interface segregation' to fix this repetitive violation. You should describe what you do to remove repeated code that is 'common' for each of these classes . Diagrams are ok but keep them inside the square given below. **(10pts)**

Single repunsibility
OOen/closed
LisKov substitution
Interface segregtion
Dependancy inversion

The problamatic code above both use an interface for Iworker. We don't want or need to have all the same methods and variables repeated for both SalaryWorker and ContractWorker

20) Given a table named **'customers'** has this data: Study this data very carefully.

customers | Enter a SQL expression to filter results (use Ctrl+Space)

	customer_id	first_name	last_name	email	phone_number
1	1	John	Doe	johndoe@example.com	123-456-7890
2	2	Jane	Smith	janesmith@example.com	987-654-3210
3	3	Bob	Johnson	bobjohnson@example.com	555-1212
4	4	Mary	Williams	marywilliams@example.com	444-5555
5	5	David	Brown	davidbrown@example.com	333-6666
6	6	Michael	Miller	michaelmiller@example.com	222-7777
7	7	Lisa	Jones	lisajones@example.com	111-8888
8	8	Robert	Davis	robertdavis@example.com	999-9999
9	9	Elizabeth	Garcia	elizabethgarcia@example.com	888-8888
10	10	James	Rodriguez	jamesrodriguez@example.com	777-7777
11	11	Jennifer	Wilson	jenniferwilson@example.com	666-6666
12	12	Christopher	Martinez	christophermartinez@example.c	555-5555
13	13	Maria	Anderson	mariaanderson@example.com	444-4444
14	14	Daniel	Thomas	danielthomas@example.com	333-3333
15	15	Patricia	Taylor	patriciataylor@example.com	222-2222
16	16	Charles	Moore	charlesmoore@example.com	111-1111
17	17	Matthew	Jackson	matthewjackson@example.com	000-0000
18	18	Susan	White	susanwhite@example.com	999-0000
19	19	Joseph	Thompson	josephthompson@example.com	888-0000
20	20	Sarah	Garcia	sarahgarcia@example.com	777-0000

Analyze the TWO record sets below and give each a SQL 'SELECT' statement to give the ***record set*** of data. Reminder: 'WHERE', 'ORDER BY', 'BETWEEN', and 'LIKE'. Incorrect syntax is okay. Grading will depend on the parts of the statement in their proper order.

	first_name	last_name	email
1	Christopher	Martinez	christophermartinez@example.com
2	Jane	Smith	janesmith@example.com
3	Patricia	Taylor	patriciataylor@example.com
4	Daniel	Thomas	danielthomas@example.com
5	Joseph	Thompson	josephthompson@example.com
6	Susan	White	susanwhite@example.com

Hint: look at 'last_name' **(10pts)**

ANSWER:

```
Select * last_name
From customers
Order By last_name;
```
note should be ascending

	first_name	last_name	phone_numl	customer_id
1	Matthew	Jackson	000-0000	17
2	Charles	Moore	111-1111	16
3	Patricia	Taylor	222-2222	15
4	Daniel	Thomas	333-3333	14
5	Maria	Anderson	444-4444	13
6	Sarah	Garcia	777-0000	20
7	Joseph	Thompson	888-0000	19
8	Susan	White	999-0000	18

Hint: look at 'customer_id' and 'phone_number' **(10pts)**

ANSWER:

```
Select * FROM Customers ORDER BY phone_number DESC;
```
add this, since it didn't fit on one line

where customer_id Between (13 and 20)

21) A student wants to buy merch from the GSU Bookstore's web site. They don't need an account in the "Bookstore" system to add items to a shopping cart, you can shop as guest. However, to pay with a Panther Cash account (on Panther card) you will need to put your GSU email and Panther card number at the checkout. Below are several components that will be used to check if the student has Panther Cash in their account and then the bookstore can use it to make/complete a purchase. The below UML Sequence diagram is started with the GSU student clicking on the "buy GSU hoodie" on the bookstore (not GSU) web server. Remember an 'IF something' can use the 'ALT' split block. **(10pts)**