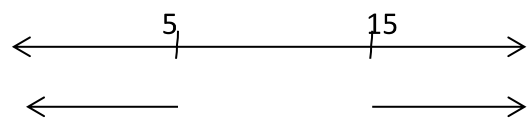
Solution:

1.1

Set equal means some cardinality with same element,

$$A = \{0\},\ \ B = \{x|x > 15\ and\ x < 5\} = \{\ \} = ø$$



C= {5},   D= {5, -5}

Not any set are equal.

1.2

Given that A is not subset of B, And B belongs to C, then A subset of C is not true.

For Example,

$$A = \{\ 0\ \},\qquad B = \{0,1\},\quad C\big\{\{0,1\},\{2\}\big\}$$

A is not a subset of C.

1.3

$$n(A \cup B) = 50,\ \ n(A) = 28,\ \ n(B) = 32,\qquad thenn(A \cap B) =?$$

$$n(A) + n(B) = \frac{n(A \cup B)}{n(A \cap B)}$$

$$n(A \cap B) = 60 - 50 = 10\quad which\ implies\ that\ n(A \cap B) = 10$$

1.4

Given that

$$A \cup B = \ A \cap B$$

$$A \cup B = \{x|x \in A\ or\ x \in B\}$$

$$A \cap B = \{x|x \in A\ and\ x \in B\}$$

$$\boldsymbol{let}\ a \in A \cup B\quad implies\ that\ a \in (A \cap B)$$

$$a \in A\ or\ a \in B\qquad |\qquad a \in A\ and\ a \in B$$

$$\mathrm{a \in A\ ,\ \ a \in B}\qquad\qquad ----\rightarrow\quad \mathrm{A\ subset\ of\ B, B\ subset\ of\ A}$$

A=B

1.5

$$P(A) \subseteq P(B)$$

P (A) is set of all subset of A

And P (B) is set of all subset of B

Set of all subset of A ⊆ set of all subset of B

A⊆ *B*

Question 2
2.1

The relation **R** on the set of integer **Z** is defined by xRy if xy>0.

Now $0 \in \mathbb{Z}$ but 0 is not related to 0 as 0.0 = 0>0 is not true. So the relation is not reflective and hence not an equivalence relation.

2.2

The relation R on the set of integer $(\mathbb{Z})$ is defined by xRy if $a^2 - b^2 \leq 3$.

$1R3$ as $1^2 - 3^2 = -8 \leq 3$ but 3 is not related to 1 as $3^2 - 1^2 = 8 > 3$.

So 1 is related to 3 but 3 is not related to 1 so the relation is not symmetric and so not an equivalence relation.

2.3
The relation r on the set of integer $(\mathbb{Z})$ is defined by $a\ r\ b$ if and only if $a + b$ is odd.

Now $1 \in \mathbb{Z}$ but 1 is not related to1 as $1 + 1 = 2$ is not odd.

So 1 is not related to 1 . So the relation is not reflexive and so not an equivalence relation.

2.4
The relation R on the set of integer $(\mathbb{Z})$ is defined by aRb if $a \neq b$

Let, aRb

$\Rightarrow a \neq b$

$\Rightarrow b \neq a$

$\Rightarrow bRa$

So if aRb then bRa and so the relation is symmetric.

$2R3$ as $2 \neq 3$

$3R2$ as $3 \neq 2$

But 2 is not related to 2 as $2 = 2$

So $2R3$ and $3R2$ but 2 is not related to 2 hence the relation is not transitive.

2.5

$A = \{1, 2, 3, 4, 5\}$

The relation R on A is defined by,

$R = \{(3,4), (5,5), (1,1), (2,2), (5,2), (1,4), (2,5), (3,1), (3,3), (4,1), (1,3), (4,3), (4,4)\}$

From this relation it can be seen that 1 is related to 1 , 4 , 3 as

$(1,1), (1,4), (1,3) \in R$. Hence, $[1] = \{1, 3, 4\}$

From this relation it can be seen that 3 is related to 1, 4, 3 as $(3,1), (3,4), (3,3) \in R$

Hence, $[3] = \{1, 3, 4\}$

Question 3

3.1

$$f: R --\rightarrow R, \qquad by\ f(x) = 2x + 3$$

a.

Find the inverse

Let

$$f(x) = y ==> x = f^{-1}(y)$$

$$2x + 3 = y$$

$$2x = y - 3 ==> x = \frac{y-3}{2}$$

$$f^{-1}(y) = \frac{y-3}{2}$$

$inverse\ of\ f^{-1}(x) = \frac{x-3}{2}$

b.

$$fof^{-1}(x) = f\left(f^{-1}(x)\right)$$

$$= f\left(\frac{x-3}{2}\right) + 3$$

$$= 2\left(\frac{x-3}{2}\right) + 3$$

$$= (x - 3) + 3 = x$$

$$f^{-1}Of(x) = f^{-1}(2x + 3)$$

$$= \left(\frac{(2x+3)-3}{2}\right) = \frac{2x}{2} = x$$

$$fOf^{-1} = f^{-1}Of$$

3.2

Let

$$f\colon R|\{2\} --\rightarrow R|\{5\}$$

$$f(x) = \frac{5x+1}{x-2}$$

For One-One

$$Let\ x1, x2 \in R|\{2\}, \qquad and\ f(x1) = f(x2)$$

Then we have to show that x1 = x2.

$$\frac{5x1+1}{x1-2} = \frac{5x2+1}{x2-2}$$

$$10(x2 - x1) + (x2 - x1) = 0$$

$$(x2 - x1)11 = 0$$

$$x2 - x1 = 0$$

$$x1 = x2$$

F is one-one

For onto:

For every y∈ $\frac{R}{\{5\}}$ $there\ exist\ an\ eliment$

$$x \in R|\{2\}\ st\ \ y = \frac{5x+1}{x-2}$$

$$xy - 2y = 5x + 1$$

$$x(y-5) = 1 + 2y$$

$$x = \frac{1+2y}{y-5}$$

$$x \in R|\{2\}, \qquad st\ \ f(x) = \frac{5x+1}{x-2} ==> f\ is\ onto$$

F is bijective.

3.3

$$f: R\text{—}\rightarrow R,\ \ f(x) = x^3 + 1$$

For y belongs to R, there exist x∈R

$$f(x) = x^3 + 1 = y$$

$$x^3 = y - 1 ==> x = (y-1)^{1/3}$$

F is surjection.

3.4

$$f: R|\{1\}\text{—}\rightarrow R|\{1\} by\ f(x) = \frac{x+1}{x-2}$$

$$inverse\ of\ f(x),\ \ f(x) = y => x = f^{-1}(y)$$

$$y = \frac{x+1}{x-2} => xy - 2y = x + 1$$

$$x(y-1) = 1 + 2y$$

$$x = (1+2y)/(y-1)$$

$$f^{-1}(y) = \frac{1+2y}{y-1}$$

Question 4:

1_ 6185762 code

```
class MySetOperations:
  def myIsEmpty(self,A):
     if len(A):
        return False
     return True
  def myDisjoint(self,A,B):

     C= A&B
     if len(C):
        return False
     return True
  def myIntersection(self,A,B):
     return A&B
  def myUnion(self,A,B):
     return A | B
  def myDifference(self,A,B):
     return A-B
  def mySymDifference(self,A,B):
     return A^B

ob=MySetOperations()
A = {1, 2, 3, 4, 5}
B = {4, 5, 6, 7, 8}
print (ob.myIsEmpty(A))
print (ob.myDisjoint(A,B))
print (ob.myIntersection(A,B))
print (ob.myUnion(A,B))
print (ob.myDifference(A,B))
print (ob.mySymDifference(A,B))
```

Output

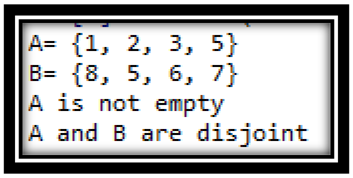
```
False
False
{4, 5}
{1, 2, 3, 4, 5, 6, 7, 8}
{1, 2, 3}
{1, 2, 3, 6, 7, 8}
```

2_ 6185762

```
def myIsEmpty(set1):
   if len(set1) == 0: # length of set is empty then return true otherwise false
      return True
   else:
```

```
        return False
def myIntersection(set1,set2):
    result=set()
    for i in set1:
        if i in set2:
            result.add(i) #adding elements to set using add()
            return result
        # u can use set1 & set2 to return intersection of two sets
        #result = set1 & set2
        # u can also use intersection()
        # result=set1.intersection(set2)
        return result
def myUnion(set1,set2):
    for i in set2:
        set1.add(i)#adding elements of set2 to set1
        return set1
    # u can use set1 | set2 to return union of two sets
    #result = set1 | set2
    # u can also use union()
    # result=set1.union(set2)
def myDisjoint(set1,set2):
    for i in set1:
        if i in set2:
            return False
        return True
    # u can also use isdisjoint()
    # result=set1.isdisjoint(s2)
s1=set([2,3,1,5])# initializing two sets using set()
s2=set([5,6,7,8])
print ("A=",s1)
print ("B=",s2)
status=myIsEmpty(s1)
if status==True:
    print ("A is empty")
else:
    print ("A is not empty")
status=myDisjoint(s1,s2)
if status==True:
    print ("A and B are disjoint")
else:
    print ("A and B are not disjoint")
    print ("Intersection of two sets are",myIntersection(s1,s2))
    print ("Union of two sets are ",myUnion(s1,s2))
```

Output



3_ 6185762

```
"""
@author: Sultan Alazemi
Panther Id : 6185762
"""

class MySetOperation:
    def _init_(self, set_obj):
        self.obj = set_obj

    def _str_(self):
        return str(self.obj)

    def is_empty(self):
        if len(self.obj) == 0:
            return True
        else:
            return False

    def disjoint(self, set_b):
        """
        This function check, is two sets are disjoint?
        :param set_b:
        :return:
        """
        for x in self.obj:  # iterating through  first set
            if x in set_b:  # checking the value  is in set b or not
                return True
        return False

    def intersection(self, set_b):
        """
        This function return intersection of two sets (comman element)
        :param set_b:
        :return:
        """
        new_set = set()
        for x in self.obj:  # iterating through  first set
            if x in set_b:  # checking the value  is in set b or not
                new_set.add(x)
        return new_set
```

```
    def union(self, set_b):
        """
        This function return union  of two sets  as new set
        :param set_b:
        :return:
        """
        new_set = set()
        for x in self.obj:
            new_set.add(x)

        for x in set_b:
            new_set.add(x)
        return new_set

    def difference(self, set_b):
        """
        This function will give  A difference B means (A-B)
        :param set_b:
        :return:
        """
        new_set = set()
        for x in self.obj:
            if x not in set_b:
                new_set.add(x)
        return new_set

    def symmetric_difference(self, set_b):
        """
        THis function will return symmetric difference between two sets
        :param set_b:
        :return:
        """

        new_set = set()
        for x in self.obj:
            if x not in set_b:
                new_set.add(x)

        for x in set_b:
            if x not in self.obj:
                new_set.add(x)
        return new_set


A = MySetOperation({1, 2, 3, 4})
C = MySetOperation(set())
B = {5, 6, 1, 7, 2, 9}
print("Set A:", A)
```

```
print("Set B:", B)
print("Set C:", C)
print("IS set C is empty:", C.is_empty())
print("Is A and B is disjoint:", A.disjoint(B))
print("A intersection B:", A.intersection(B))
print("A union B:", A.union(B))
print("A difference B:", A.difference(B))
print("A symmetric difference B:", A.symmetric_difference(B))
```

Output

```
Set A: {1, 2, 3, 4}
Set B: {1, 2, 5, 6, 7, 9}
Set C: set()
IS set C is empty: True
Is A and B is disjoint: True
A intersection B: {1, 2}
A union B: {1, 2, 3, 4, 5, 6, 7, 9}
A difference B: {3, 4}
A symmetric difference B: {3, 4, 5, 6, 7, 9}
```