https://www.python-course.eu/python3

Odd Semester (2020)

BINUS UNIVERSITY
BINUS INTERNATIONAL

Assignment Cover Letter

(Individual Work)

Student Information:	**Surname**	**Given Names**	**Student ID Number**
1.	**Kasdi**	**Handy**	**2101693504**

Course Code	**: COMP6335**	**Course Name**	**: Introduction to Programming**
Class	**: L1AC**	**Name of Lecturer(s)**	: 1. Minaldi Loeis 2. Jude Joseph Lamug Martinez
Major	**: CS**		
Title of Assignment (if any)	: Black Jack		
Type of Assignment	**: Final Project**		
Submission Pattern			
Due Date	**: 7-11-2017**	**Submission Date**	**: 6-11-2017**

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer's instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

Plagiarism/Cheating

BiNus International seriously regards all forms of plagiarism, cheating and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

Declaration of Originality

By signing this assignment, Iunderstand, accept and consent to BiNus International terms and policy on plagiarism. Herewith Ideclare that the work contained in this assignment is myown work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student: (Name of Student)

“BlackJack3”

Name :Handy Fernando Kasdi

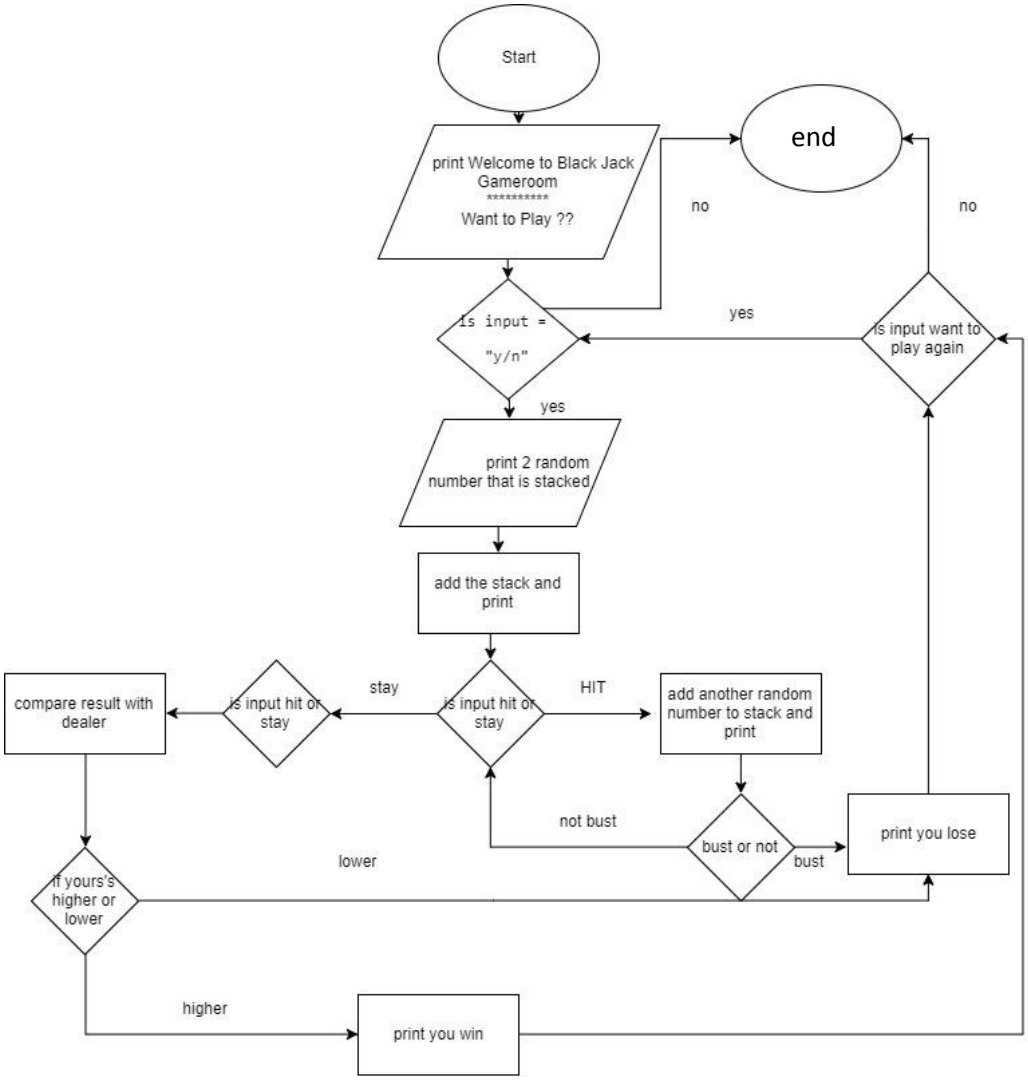
ID :2101693504

I. Description

The function of this program:

This program is meant to help people gain more stress by play a game of black jack without using money because gambling is frobidden. And it’s help developer and costumer to enjoy themself playing card when feeling lonely.

II.a. Design/Plan



https://www.python-course.eu/python3

II.b. Explanation of Each function

UML DIAGRAM

Hand
-hand
+add card +bust +reset hand

Card
-rank -suit -ranks `[None,"Ace", 2, 3, 4, 5, 6, 7, 8, 9, 10, "Jack", "Queen", "King"]` -suits `{"d": "Diamonds", "c": "Clubs", "h": "Hearts", "s": "Spades"}`
+get rank +get suit

Class diagram:

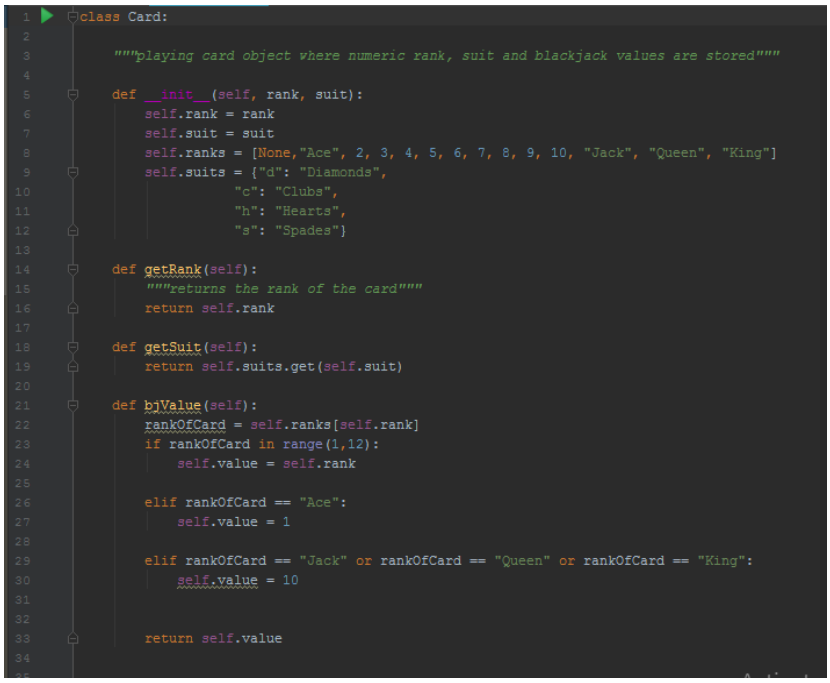
1. **Classes**
   - provide a means of bundling data and functionality together.
   - class creates a new *type* of object, allowing new *instances* of that type to be made.
   - Classes in "Black Jack Project" define the ranks, the cards, and type of cards
2. **Def Function**
   - reusable code that is used to perform a single, related action.
   - provide better modularity for your application and a high degree of code reusing.
   - It's run almost all the program in one call.
3. **Print**
   - Function that display strings and integer on the output
4. **Input**
   - Input is usually a string that is used as a choice to make by user that use the program
5. **While loop**
   - are used for repeating sections of code.

- unlike a *for* loop, the *while* loop will not run *n* times, but until a defined condition is no longer met. If the condition is initially false,

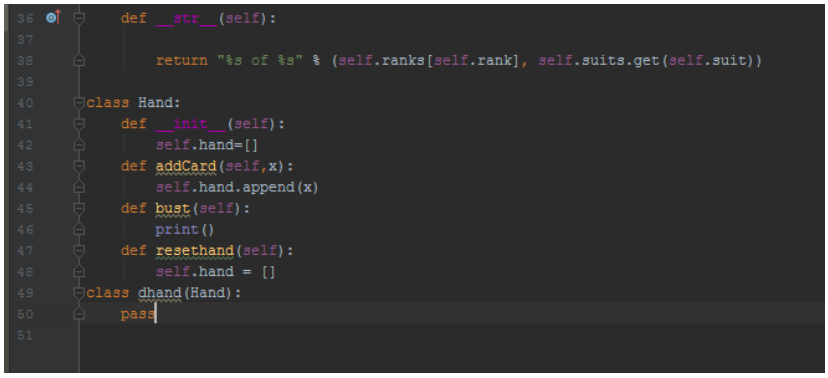
6. **If function**
    - To determind a condition weather it's true or false
    - Usually use to run a new code with conditions
7. **Elif**
    - To run another condition when 1st condition is false
    - Could be use as many times as you want
8. **Else**
    - To run the program when all the condition is false

Source code:

Class diagram:



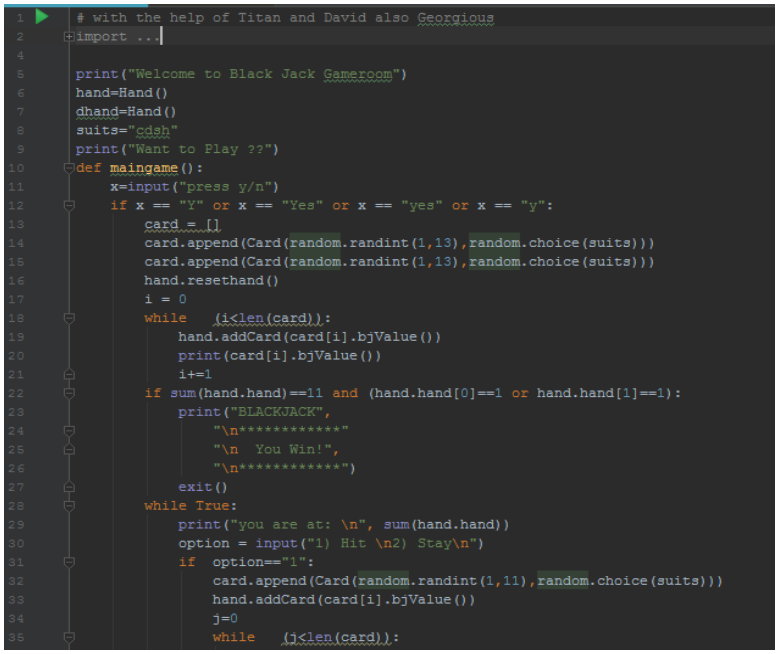
```
class Card:

    """playing card object where numeric rank, suit and blackjack values are stored"""

    def __init__(self, rank, suit):
        self.rank = rank
        self.suit = suit
        self.ranks = [None,"Ace", 2, 3, 4, 5, 6, 7, 8, 9, 10, "Jack", "Queen", "King"]
        self.suits = {"d": "Diamonds",
                      "c": "Clubs",
                      "h": "Hearts",
                      "s": "Spades"}

    def getRank(self):
        """returns the rank of the card"""
        return self.rank

    def getSuit(self):
        return self.suits.get(self.suit)

    def bjValue(self):
        rankOfCard = self.ranks[self.rank]
        if rankOfCard in range(1,12):
            self.value = self.rank

        elif rankOfCard == "Ace":
            self.value = 1

        elif rankOfCard == "Jack" or rankOfCard == "Queen" or rankOfCard == "King":
            self.value = 10


        return self.value
```



```
    def __str__(self):

        return "%s of %s" % (self.ranks[self.rank], self.suits.get(self.suit))

class Hand:
    def __init__(self):
        self.hand=[]
    def addCard(self,x):
        self.hand.append(x)
    def bust(self):
        print()
    def resethand(self):
        self.hand = []
class dhand(Hand):
    pass
```

https://www.python-course.eu/python3

Main program:



```
# with the help of Titan and David also Georgious
import ...

print("Welcome to Black Jack Gameroom")
hand=Hand()
dhand=Hand()
suits="cdsh"
print("Want to Play ??")
def maingame():
    x=input("press y/n")
    if x == "Y" or x == "Yes" or x == "yes" or x == "y":
        card = []
        card.append(Card(random.randint(1,13),random.choice(suits)))
        card.append(Card(random.randint(1,13),random.choice(suits)))
        hand.resethand()
        i = 0
        while   (i<len(card)):
            hand.addCard(card[i].bjValue())
            print(card[i].bjValue())
            i+=1
        if sum(hand.hand)==11 and (hand.hand[0]==1 or hand.hand[1]==1):
            print("BLACKJACK",
                "\n***********"
                "\n  You Win!",
                "\n***********")
            exit()
        while True:
            print("you are at: \n", sum(hand.hand))
            option = input("1) Hit \n2) Stay\n")
            if  option=="1":
                card.append(Card(random.randint(1,11),random.choice(suits)))
                hand.addCard(card[i].bjValue())
                j=0
                while   (j<len(card)):
```

```
                j=0
                while   (j<len(card)):
                    print(card[j].bjValue())
                    j+=1
                if  (sum(hand.hand)>21):
                    print("You lose sucker")
                    break

            if  option=="2":
                while   (sum(dhand.hand)<17):
                        dhand.addCard(Card(random.randint(1,11),random.choice(suits)).bjValue())
                print("Dealer has: \n",sum(dhand.hand))
                if  (sum(dhand.hand)>21):
                    print("Dealer Bust\nYou win")
                    break
                elif(sum(hand.hand)>sum(dhand.hand) and sum(hand.hand)<=21):
                    print("You win")
                    break
                elif(sum(hand.hand)==sum(dhand.hand)):
                    print("Draw")
                    break
                else:
                    print("You lose")
                    break
    else:
        exit()

while True:
    maingame()
    print("want to play again?")
```