COSC 320 Priciples of Programming

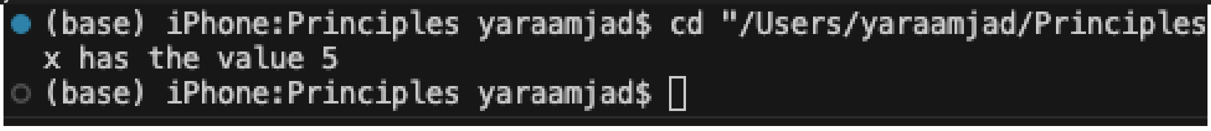
Lab 1

Joud Herzallah - 100061269

## 01 - Variable:

Q1)

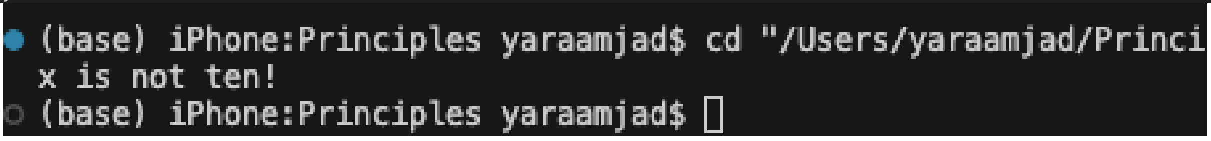
```
fn main() {
        // TODO: Add the missing keyword.
       let x = 5;
        println!("x has the value {x}");

}
```



```
● (base) iPhone:Principles yaraamjad$ cd "/Users/yaraamjad/Principles
  x has the value 5
○ (base) iPhone:Principles yaraamjad$ 
```

Q2)

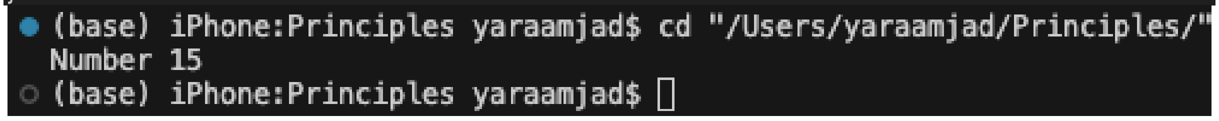
```
fn main() {
    // TODO: Change the line below to fix the compiler error.
    let x: u8=5;

    if x == 10 {
        println!("x is ten!");
    } else {
        println!("x is not ten!");
    }
}
```



```
● (base) iPhone:Principles yaraamjad$ cd "/Users/yaraamjad/Princi
  x is not ten!
○ (base) iPhone:Principles yaraamjad$ 
```

Q3)

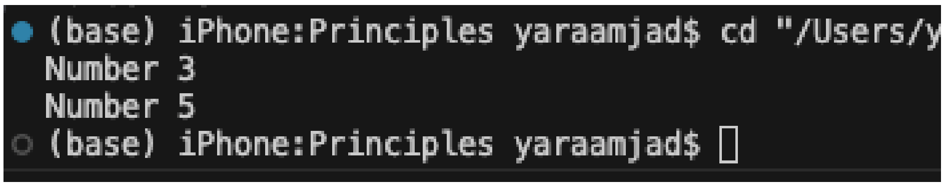
```
fn main() {
    // TODO: Change the line below to fix the compiler error.
    let x: i32 =15;
    println!("Number {x}");
}
```



```
● (base) iPhone:Principles yaraamjad$ cd "/Users/yaraamjad/Principles/"
  Number 15
○ (base) iPhone:Principles yaraamjad$ ▯
```

Q4)

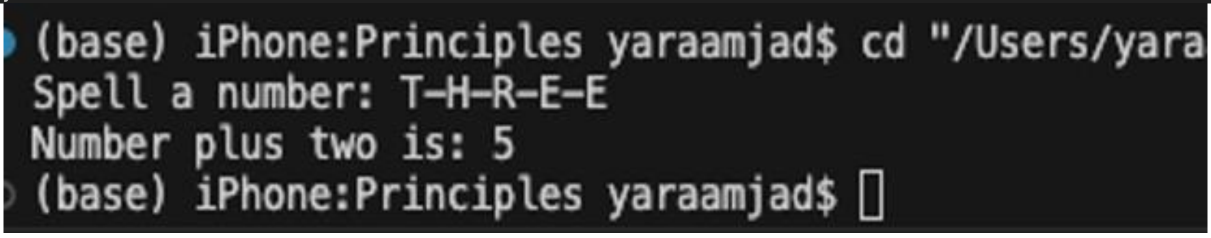
```
// TODO: Fix the compiler error.
fn main() {
    let mut x = 3;
    println!("Number {x}");

    x = 5; // Don't change this line
    println!("Number {x}");
}
```



```
● (base) iPhone:Principles yaraamjad$ cd "/Users/y
  Number 3
  Number 5
○ (base) iPhone:Principles yaraamjad$ ▯
```

Q5)

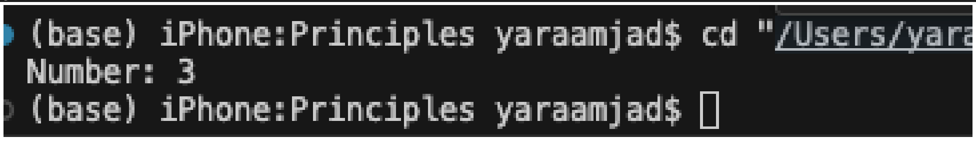
```
fn main() {
    let number = "T-H-R-E-E"; // Don't change this line
    println!("Spell a number: {}", number);

    // TODO: Fix the compiler error by changing the line below without renaming the variable.
    let number : i32 = 3;
    println!("Number plus two is: {}", number + 2);
}
```



```
(base) iPhone:Principles yaraamjad$ cd "/Users/yara
Spell a number: T-H-R-E-E
Number plus two is: 5
(base) iPhone:Principles yaraamjad$ ▯
```

Q6)

```
// TODO: Change the line below to fix the compiler error.
const NUMBER :i32 = 3;

fn main() {
    println!("Number: {NUMBER}");
}
```

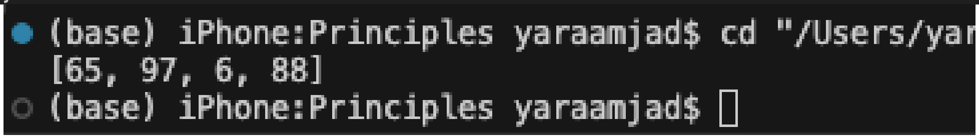


```
(base) iPhone:Principles yaraamjad$ cd "/Users/yara
Number: 3
(base) iPhone:Principles yaraamjad$ ▯
```

## 06 – Move Semantics:

Q1)

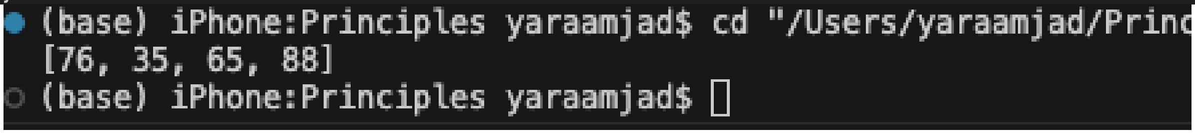
```
// TODO: Fix the compiler error in this function.
fn fill_vec( vec: Vec<i32>) -> Vec<i32> {
    let mut vec = vec;

    vec.push(88);

    vec
}

fn main() {

    let vec0 = vec![65, 97, 6];
    let vec1 = fill_vec(vec0);
    println!("{:?}", vec1);
}

#[cfg(test)]
mod tests {
    use super::*;

    #[test]
    fn move_semantics1() {
        let vec0 = vec![22, 44, 66];
        let vec1 = fill_vec(vec0);
        assert_eq!(vec1, vec![22, 44, 66, 88]);
    }
}
```



```
● (base) iPhone:Principles yaraamjad$ cd "/Users/yar
  [65, 97, 6, 88]
○ (base) iPhone:Principles yaraamjad$ 
```

Q2)

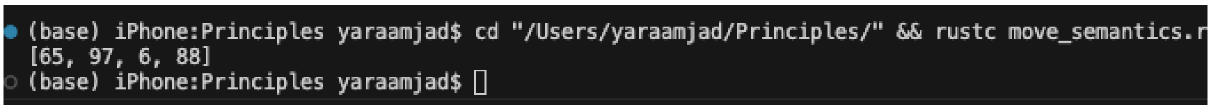
```

fn fill_vec(vec: Vec<i32>) -> Vec<i32> {
    let mut vec = vec;

    vec.push(88);

    vec
}

fn main() {
    let vec0 = vec![76, 35, 65];

    let vec1 = fill_vec(vec0.clone());
    println!("{:?}", vec1);
}

#[cfg(test)]
mod tests {
    use super::*;
    // TODO: Make both vectors `vec0` and `vec1` accessible at
the same time to
    // fix the compiler error in the test.
    #[test]
    fn move_semantics2() {
        let vec0 = vec![22, 44, 66];

        let vec1 = fill_vec(vec0.clone());


        assert_eq!(vec0, [22, 44, 66]);
        assert_eq!(vec1, [22, 44, 66, 88]);
    }
}
```



```
● (base) iPhone:Principles yaraamjad$ cd "/Users/yaraamjad/Princ
  [76, 35, 65, 88]
○ (base) iPhone:Principles yaraamjad$ ▯
```

Q3)

```
// TODO: Fix the compiler error in the function without adding any new line.
fn fill_vec(mut vec: Vec<i32>) -> Vec<i32> {
    vec.push(88);

    vec
}

fn main() {
    // You can optionally experiment here.
    let vec0 = vec![65, 97, 6];
    let vec1 = fill_vec(vec0);
    println!("{:?}", vec1);
}

#[cfg(test)]
mod tests {
    use super::*;

    #[test]
    fn move_semantics3() {
        let vec0 = vec![22, 44, 66];
        let vec1 = fill_vec(vec0);
        assert_eq!(vec1, [22, 44, 66, 88]);
    }
}
```



```
● (base) iPhone:Principles yaraamjad$ cd "/Users/yaraamjad/Principles/" && rustc move_semantics.r
  [65, 97, 6, 88]
○ (base) iPhone:Principles yaraamjad$ ▯
```